

# Prediction with a Short Memory

**Sham Kakade**

University of Washington  
sham@cs.washington.edu

**Percy Liang**

Stanford University  
pliang@cs.stanford.edu

**Vatsal Sharan**

Stanford University  
vsharan@stanford.edu

**Gregory Valiant**

Stanford University  
valiant@stanford.edu

## Abstract

We consider the problem of predicting the next observation given a sequence of past observations. We show that for any distribution over observations, if the mutual information between past observations and future observations is upper bounded by  $I$ , then a simple Markov model over the most recent  $I/\epsilon$  observations can obtain KL error  $\epsilon$  with respect to the optimal predictor with access to the entire past. For a Hidden Markov Model with  $n$  states,  $I$  is bounded by  $\log n$ , a quantity that does not depend on the mixing time. We also demonstrate that the simple Markov model cannot really be improved upon: First, a window length of  $I/\epsilon$  ( $I/\epsilon^2$ ) is information-theoretically necessary for KL error ( $\ell_1$  error). Second, the  $d^{\Theta(I/\epsilon)}$  samples required to accurately estimate the Markov model when observations are drawn from an alphabet of size  $d$  is in fact necessary for any computationally tractable algorithm, assuming the hardness of strongly refuting a certain class of CSPs.

# 1 Memory, Modeling, and Prediction

We consider the problem of predicting the next observation  $x_t$  given a sequence  $x_1, x_2, \dots, x_{t-1}$  of past observations, which could have complex dependencies. This *sequential prediction* setting arises naturally in natural language modeling [1], where the goal is to predict the next word in a document given the previous words; other applications include speech synthesis [2] and financial forecasting. The abstract problem has received much attention over the last half century from multiple communities including TCS, machine learning, and coding theory. The fundamental challenge is the same: *What do we remember about the past that enables effective prediction of the future?*

A simple baseline for sequential prediction is the  $\ell$ -th order Markov model, which specifies the distribution over  $x_t$  given the past  $\ell - 1$  observations,  $x_{t-\ell+1}, \dots, x_{t-1}$ . While this naive model cannot remember anything about the past farther than  $\ell$  time steps ago, the Markov model has nonetheless proven to be surprisingly difficult to beat. Its algorithmic and statistical simplicity permits scaling to large datasets, and with proper smoothing [1], it has been an crucial component in state-of-the-art machine translation and speech recognition systems. In the last five years, however, in the midst of the deep learning revolution, recurrent neural networks [3], in particular, Long Short-Term Memory (LSTM) networks [4, 5], have finally demonstrated empirical gains over Markov-like models [6, 7]. RNNs encode the past  $x_1, \dots, x_{t-1}$  as a real vector  $h_t$ , which offers the ability to capture long-range dependencies. On the other hand, Hidden Markov Models (HMMs), which also capture long-range dependencies, have fared less well, and the nature of these gaps are not well-understood.

Despite the long history of sequential prediction and its importance in applications, many fundamental questions remain: (i) How much memory is necessary to accurately predict future observations, and what properties of the underlying process determine this requirement? (ii) Must one remember significant information about the distant past or is a short-term memory (as in the Markov model) sufficient? (iii) What data distributions permit computationally efficient prediction algorithms?

**Upper bounds.** We begin by showing the following elementary proposition, which addresses the first two questions:

**Proposition 1.** *Let  $\mathcal{M}$  be any distribution over sequences, with mutual information  $I(\mathcal{M})$  between the past observations  $\dots, x_{t-2}, x_{t-1}$  and future observations  $x_t, x_{t+1}, \dots$ . The best  $\ell$ -th order Markov model obtains average KL error  $I(\mathcal{M})/\ell$  for predicting the distribution of the next observation, with respect to the true conditional distribution of  $x_t$  given all past observations.*

There are three points worth emphasizing. First, Proposition 1 shows that a Markov model can predict accurately on *any* data-generating distribution (provided the order of the Markov model scales with the complexity of the distribution). At a time where increasingly complex models such as recurrent neural networks and neural Turing machines [8] are in vogue, Proposition 1 serves as a baseline theoretical result. After all, the Markov model is algorithmically trivial, in contrast with the computational difficulty of parameter estimation in the other models.

Second, to obtain accurate prediction *on average*, it suffices to have short-term memory, as the Markov model makes predictions based on a window of the  $\ell$  most recent observations. This result is perhaps somewhat surprising given that the true distribution could have arbitrarily long-range dependence. The intuition why short-term memory suffices is that when the Markov model makes a prediction for  $x_t$ , it either makes a correct prediction, or if it makes an incorrect prediction, then  $x_t$  contains some significant amount of new information about the past history, which will be of use when predicting  $x_{t+1}$ .

Third, the average error scales with the mutual information  $I(\mathcal{M})$  between the past and future. While the mutual information seems intuitive, the fact that it is the right quantity is subtle. Consider the following setting: Given a joint distribution over random variables  $A$  and  $B$ , suppose we wish to define a function  $f$  that maps  $A$  to a binary “advice” string  $f(A)$  (possibly of variable length) such that  $B$  only depends on  $A$  through  $f(A)$ . In our setting,  $A$  is the sequence of past observations,  $B$  is the sequence of future observations, and  $f(A)$  is the information about the past sufficient to predict the future. As is shown in Harsha et al. [9], there are joint distributions over  $(A, B)$  such that even on average, the minimum length of the advice string necessary for the above task is exponential in the mutual information  $I(A; B)$ .<sup>1</sup> Given the fact that this mutual information is not even an upper bound on the amount of memory that an optimal algorithm (computationally unbounded, and with complete knowledge of the distribution) would require, Proposition 1 might be surprising.

Proposition 1, framed in terms of the mutual information of the past and future, has immediate implications for a number of well-studied models of sequential data such as Hidden Markov Models (HMMs):

**Corollary 1.** *Suppose observations were generated from an underlying HMM  $\mathcal{M}$  with  $n$  hidden states. Then the  $\ell$ -th order Markov model obtains  $(\log n)/\ell$  average KL error with respect to the optimal predictor that knows the underlying HMM and has access to all past observations.*

The above corollary shows that the required window length is independent of the mixing time of the underlying Markov chain. For quickly mixing chains, short windows are obviously sufficient as the process itself forgets the distant past. For Markov chains with large mixing time or that do not mix at all, the intuition behind the plausibility of the above result is the following: Either short windows are relatively “rich” and contain significant amounts of information about the future, or short windows are relatively “boring” and contain little information about the future; in the latter case, however, accurate prediction on average is also relatively easy. We describe this intuition in greater detail at the beginning of Section 2 before giving our proof of Proposition 1.

**Lower bounds.** If the observations take on values from an alphabet of size  $d$ , then the naive  $\ell$ -th order Markov model has  $d^\ell$  parameters, which becomes expensive to estimate for large  $d$ , which occurs in naturally language processing where observations are words. We show that this complexity is in some sense inevitable, assuming that the problem of strongly refuting a certain class of CSPs is hard, which was conjectured in [10] and studied in related works [11] and [12]. See Section 3 for a description of this class and discussion of the conjectured hardness.

**Theorem 1.** *Assuming the hardness of strongly refuting a certain class of CSPs, for all integers  $t > 0$  and  $0 < \epsilon \leq 0.1$ , there exists a family of distributions over sequences with observations drawn from an alphabet of size  $d$ , such that every distribution  $\mathcal{M}$  in the family has mutual information  $I(\mathcal{M}) \in [ct, t]$  for some fixed constant  $c$ , but any polynomial time algorithm that achieves average error  $\epsilon$  for a random distribution in the family requires  $d^{\Theta(I(\mathcal{M})/\epsilon)}$  samples from  $\mathcal{M}$ .*

A different but equally relevant regime is where the alphabet size  $d$  is small compared to the scale of dependencies in the sequence (for example, when predicting characters [13]). We show

---

<sup>1</sup>It is worth noting that if the “advice” string  $s$  is sampled first, and then  $A$  and  $B$  are defined to be random functions of  $s$ , then the length of  $s$  can be related to  $I(A; B)$  (see [9]). This can also be interpreted as a two-player communication game where one player generates  $A$  and the other generates  $B$  given limited communication. This latter setting where  $s$  is generated first corresponds to allowing shared randomness; such a setting is, however, not relevant to the sequential prediction setting we consider.

lower bounds in this regime of the same flavor as those of Theorem 1 except based on the problem of learning a noisy parity function; the (very slightly) subexponential algorithm of Blum et al. [14] for this task means that we lose at least a superconstant factor in the exponent in comparison to the positive results of Proposition 1.

**Proposition 2.** *Let  $f(n)$  denote a lower bound on the amount of time and samples required to learn parity with noise on uniformly random  $n$ -bit inputs. For all sufficiently large  $t$  and  $0 < \epsilon \leq 0.1$ , there exists a family of distributions over sequences of binary strings, such that every distribution  $\mathcal{M}$  in the family satisfies  $I(\mathcal{M}) \in [ct, t]$  for some fixed constant  $c$ , and any algorithm that achieves average prediction error  $\epsilon$  for a random distribution in the family requires at least  $f(I(\mathcal{M})/\epsilon)$  time or samples.*

The above bounds show conditional computational hardness of improving significantly on the sample requirements of the naive  $\ell$ -th order Markov model. One natural question is whether it is possible for any algorithm (even computationally unconstrained) to achieve better average prediction based only on short windows (of length  $\ell$ ). The following information-theoretic lower bound shows that even for the class of HMMs, up to a constant factor, no algorithm based on windows of this length can beat the Markov model. This optimality of the naive Markov model holds both for KL-divergence and for  $\ell_1$  distance. Proposition 1 via Pinsker’s inequality shows that an  $\ell_1$  error of  $\epsilon$  can be achieved with windows of length  $\ell = I(\mathcal{M})/\epsilon^2$ . The following proposition shows that even a computationally unbounded prediction algorithm, which has knowledge of the HMM underlying the observations, cannot improve on this  $\ell_1$  prediction error beyond a constant factor.

**Proposition 3.** *There is an absolute constant  $c < 1$  such that for all  $0 < \epsilon < 0.5$  and sufficiently large  $n$ , there exists an HMM with  $n$  hidden states such that it is not information-theoretically possible to obtain  $\ell_1$  average prediction error less than  $\epsilon$  using windows of only length  $c \log n / \epsilon^2$ .*

**A Negative Perspective On Average Error.** Proposition 1 shows that, given sufficient data, a trivial model that only looks at the most recent window of observations can be reasonably successful at prediction, even though such a model clearly fails to capture any long-range dependencies or structure of the data.<sup>2</sup> Such long-range dependencies seems important for understanding natural language—indeed, the main message of a narrative is not conveyed in any single short segment. More generally, higher-level intelligence seems to be about the ability to judiciously decide what aspects of the observation sequence are worth remembering and updating a model of the world based on these aspects.

Thus, Proposition 1, despite being nominally a positive result on prediction, should in fact be interpreted as a negative result—that average error is not a good metric. Indeed, average error dishes out too much reward to models that simply coast through life: a Markov model can drive down its error by simply predicting function words such as “the”. A more desirable metric perhaps should have more of a stringent worst-case flavor. Finding a good metric that can drive progress forward in a meaningful direction remains an important open problem.

---

<sup>2</sup>Many natural language generation systems also suffer from this issue. One amusing example is the recent sci-fi short film *Sunspring* whose script was automatically generated by an LSTM. Locally, each sentence of the dialogue (mostly) makes sense, though there is no cohesion over longer time frames, and certainly no overarching plot trajectory (despite the brilliant acting).

## 1.1 Related Work

**Sequential Prediction in Practice.** This work was initiated by the desire to understand the role of memory in sequential prediction, and the belief that modeling long-range dependencies is important for complex tasks such as understanding natural language. Recurrent neural networks are known to lack long-term memory if they are to be stable [3], while the Long Short-Term Memory (LSTM) [4] has been hugely successful as the building block for various other models in deep learning, including attention-based models [15] and neural Turing machines [8]. Despite the empirical success and the fact that memory is made explicit, theoretical understanding is still lacking. A closer inspection at an LSTM reveals that it too will forget the past exponentially eventually if it is to be stable. To gain more insight into the problem, we started by analyzing a simple Markov model, and found to our surprise that it performed as well as one could hope.

**Parameter Estimation.** It is interesting to compare using a Markov model with methods that attempt to learn an underlying model. For example, method of moments algorithms [16, 17] allow one to estimate a certain class of hidden Markov model with polynomial sample and computational complexity. These ideas have been extended to learning neural networks [18] and input-output RNNs [19]. Using different methods, Arora et al. [20] showed how to learn certain random deep neural networks. Learning the model directly can result in better sample efficiency, and also provide insights into the structure of the data. Of course, a major challenge is that these methods require the true data-generating distribution to be in the model family that we are learning, which is a very strong assumption in practice.

**Universal Prediction and Coding Theory.** On the other end of the spectrum is the class of no-regret online learning methods which assume that the data generating distribution can even be adversarial [21]. However, the nature of these results are fundamentally different from ours: whereas we are comparing to the perfect model that can look at the infinite past, online learning methods typically compare to a fixed set of experts, which is typically much weaker.

There is much work on sequential prediction based on KL-error from the information theory and statistics communities. The philosophy of these approaches are often more adversarial, with perspectives ranging from minimum description length [22, 23] and individual sequence settings [24], where no model of the data distribution process is assumed. With regards to worst case guarantees (where there is no data generation process) and the notion of optimality is *regret*, there is a line of work on both minimax rates and the performance of Bayesian algorithms, the latter of which has favorable guarantees in a sequential setting. With regards to minimax rates, [25] provides an exact characterization of the minimax strategy, though the applicability of this approach is often limited to settings where the strategies available to the learner is relatively small (i.e., the normalizing constant in [25] must exist). More generally, there has been considerable work on the regret in information-theoretic and statistical settings, such as the works in [26, 24, 27, 28, 29, 30, 31, 32].

With regards to log-loss more broadly, there is considerable work on information consistency (convergence in distribution) and minimax rates with regards to statistical estimation in parametric and non-parametric families [33, 34, 35, 36, 37, 38]. In some of these settings, e.g. minimax risk in parametric, i.i.d. settings, there are characterizations in terms of mutual information [34].

There is also work on universal lossless data compression algorithm, such as the celebrated Lempel-Ziv algorithm [39]. Here, the setting is rather different as it is one of coding the entire sequence (in a block setting) rather than prediction loss.

Finally, we already discussed the very related work of Harsha et al. [9], which considers generating a sample from a joint distribution over random variables  $(A, B)$  via the following communication scheme: the sender samples  $A$  and transmits some code  $f(A)$ , while the receiver samples  $B$  conditioned on  $f(A)$ . In the setting where the two parties have shared randomness (which is

not applicable to our sequential prediction setting), the expected number of bits in  $f(A)$  lies in  $[I(A; B), 2I(A; B)]$ . Without shared randomness, the communication cost can be exponential in  $I(X; Y)$ .

## 1.2 Definitions and Notation

For any random variable  $X$ , we denote its distribution as  $P(X)$ . The mutual information between two random variables  $X$  and  $Y$  is defined as  $I(X; Y) = H(Y) - H(Y|X)$  where  $H(Y)$  is the entropy of  $Y$  and  $H(Y|X)$  is the conditional entropy of  $Y$ . The conditional mutual information  $I(X; Y|Z)$  is defined as:

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z) = \mathbb{E}_{x,y,z} \log \frac{P(X|Y, Z)}{P(X|Z)} = \mathbb{E}_{y,z} D_{KL}(P(X|Y, Z) \parallel P(X|Z))$$

where  $D_{KL}(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$  is the KL divergence. Note that we are slightly abusing notation here as  $D_{KL}(P(X|Y, Z) \parallel P(X|Z))$  should technically be  $D_{KL}(P(X|Y = y, Z = z) \parallel P(X|Z = z))$ . But we will ignore the assignment in the conditioning when it is clear from the context. Mutual information obeys the following chain rule:  $I(X_1, X_2; Y) = I(X_1; Y) + I(X_2; Y|X_1)$ .

Given a distribution over infinite sequences,  $\{x_t\}$  generated by some model  $\mathcal{M}$  where  $x_t$  is random variable denoting the output at time  $t$ , we will use the shorthand  $x_i^j$  to denote the random variable for the subsequence of outputs  $\{x_i, \dots, x_j\}$ . The distribution of  $\{x_t\}$  is *stationary* if the joint distribution of any subset of the sequence of random variables  $\{x_t\}$  is invariant with respect to shifts in the time index. Hence  $P(x_{i_1}, x_{i_2}, \dots, x_{i_n}) = P(x_{i_1+l}, x_{i_2+l}, \dots, x_{i_n+l})$  for any  $l$  if the process is stationary.

In this work, we are interested in studying how well the output  $x_t$  can be predicted by an algorithm which only looks at the past  $(\ell - 1)$  outputs. The predictor  $\mathcal{A}_\ell$  maps a sequence of  $(\ell - 1)$  observations to a predicted distribution of the next observation. We denote the predictive distribution of  $\mathcal{A}_\ell$  at time  $t$  as  $Q_{\mathcal{A}_\ell}(x_t|x_{t-\ell+1}^{t-1})$ . We refer to the Bayes optimal predictor using only windows of length  $\ell$  as  $\mathcal{P}_\ell$ , hence the prediction of  $\mathcal{P}$  at time  $t$  is  $P(x_t|x_{t-\ell+1}^{t-1})$ .  $\mathcal{P}_\ell$  is just the naive  $\ell$ -th order Markov predictor provided with the true distribution of the data. Let the Bayes optimal predictor looking at the entire history of the model be  $\mathcal{P}_\infty$ , the prediction of  $\mathcal{P}_\infty$  at time  $t$  is  $P(x_t|x_{-\infty}^{t-1})$ . We will evaluate the predictions of  $\mathcal{A}_\ell$  and  $\mathcal{P}_\ell$  with respect to  $\mathcal{P}_\infty$  over a long time window  $[0 : T - 1]$ .

The crucial property of the distribution that is relevant to our results is the mutual information between past and future observations. For a stochastic process  $\{x_t\}$  generated by some model  $\mathcal{M}$  we define the mutual information  $I(\mathcal{M})$  of the model  $\mathcal{M}$  as the mutual information between the past and future, averaged over the window  $[0 : T - 1]$ .

$$I(\mathcal{M}) = \frac{1}{T} \sum_{t=0}^{T-1} I(x_t^\infty; x_{-\infty}^{t-1}) \quad (1.1)$$

If the process  $\{x_t\}$  is stationary, then  $I(x_t^\infty; x_{-\infty}^{t-1})$  is the same for all time steps hence  $I(\mathcal{M}) = I(x_0^\infty; x_{-\infty}^{-1})$ . If the average does not converge, we can define  $I(\mathcal{M}, [0 : T - 1])$  as the mutual information for the window  $[0 : T - 1]$ , and the results hold true with  $I(\mathcal{M})$  replaced by  $I(\mathcal{M}, [0 : T - 1])$ .

We compare the prediction of the predictor  $\mathcal{P}_\ell$  and  $\mathcal{A}_\ell$  with respect to  $\mathcal{P}_\infty$ . Let  $F(P, Q)$  be some measure of distance between two predictive distributions. In this work, we consider the KL-divergence,  $\ell_1$  distance and the relative zero-one loss between the two distributions. The KL-divergence and  $\ell_1$  distance between two distributions are defined in the standard way. We define

the relative zero-one loss as the difference between the zero-one loss of the optimal predictor  $\mathcal{P}_\infty$  and the algorithm  $\mathcal{A}_\ell$ . We define the expected loss of any predictor  $\mathcal{A}_\ell$  with respect to the optimal predictor  $\mathcal{P}_\infty$  and a loss function  $F$  as follows:

$$\begin{aligned}\delta_F^{(t)}(\mathcal{A}_\ell) &= \mathbb{E}_{x_{-\infty}^{t-1}} \left[ F(P(x_t|x_{-\infty}^{t-1}), Q_{\mathcal{A}_\ell}(x_t|x_{t-\ell+1}^{t-1})) \right] \\ \delta_F(\mathcal{A}_\ell) &= \frac{1}{T} \sum_{t=0}^{T-1} \delta_F^{(t)}(\mathcal{A}_\ell)\end{aligned}$$

We also define  $\hat{\delta}_F^{(t)}(\mathcal{A}_\ell)$  and  $\hat{\delta}_F(\mathcal{A}_\ell)$  for the algorithm  $\mathcal{A}_\ell$  in the same fashion as the error in estimating  $P(x_t|x_{t-\ell+1}^{t-1})$ , the true conditional distribution of the model  $\mathcal{M}$ .

$$\begin{aligned}\hat{\delta}_F^{(t)}(\mathcal{A}_\ell) &= \mathbb{E}_{x_{t-\ell+1}^{t-1}} \left[ F(P(x_t|x_{t-\ell+1}^{t-1}), Q_{\mathcal{A}_\ell}(x_t|x_{t-\ell+1}^{t-1})) \right] \\ \hat{\delta}_F(\mathcal{A}_\ell) &= \frac{1}{T} \sum_{t=0}^{T-1} \hat{\delta}_F^{(t)}(\mathcal{A}_\ell)\end{aligned}$$

## 2 Predicting Well with Short Windows

We start with some intuition for why short windows should suffice to make accurate predictions. Consider a sequence of binary observations generated by an  $n$ -state HMM, and note that the Markov model with knowledge of the true distribution of the data that predicts based on windows of length  $\ell$  corresponds exactly to the Bayes optimal predictor that has knowledge of the true HMM but only observes the most recent  $\ell - 1$  observations. Consider making predictions of  $x_0, x_1, \dots$ , where the prediction at time  $i$  is the Bayes optimal prediction based on the previous window of  $i$  observations and knowledge of the true underlying HMM. One of two things can occur at each time step: either the prediction of  $x_i$  is accurate, or  $x_i$  is unexpected, in which case it provides information about the hidden state that the Markov process was in at time 0. Since there are at most  $\log n$  bits of entropy in the stationary distribution of the Hidden Markov process, by the time the predictor has made more than  $\log n$  errors, it will essentially know the hidden state at time 0 (in which case all future predictions will be accurate, as the hidden state at time 0 contains all the relevant information about the infinite history  $x_{-\infty}, \dots, x_{-1}$ ). Hence for  $i \geq c \log n$  we expect the predictor to have made only  $\approx \log n$  mistakes; this translates to the statement that the Markov model on windows of length  $\ell = c \log n$  should be expected to achieve error roughly  $1/c$ .

The above argument sketches the intuition for why, in the case where the sequence of observations is generated by a HMM, the expected prediction error of the Markov model should be small provided the window length is significantly longer than the entropy of the stationary distribution of the underlying hidden Markov chain. To establish our general proposition, which applies beyond this HMM setting, we provide an elementary and purely information theoretic proof.

**Proposition 1.** *For any data-generating distribution  $\mathcal{M}$  with mutual information  $I(\mathcal{M})$  between past and future observations, the best  $\ell$ -th order Markov model  $\mathcal{P}_\ell$  obtains average KL-error,  $\delta_{KL}(\mathcal{P}_\ell) \leq I(\mathcal{M})/\ell$  with respect to the optimal predictor with access to the infinite history. Also, any predictor  $\mathcal{A}_\ell$  with  $\hat{\delta}_{KL}(\mathcal{A}_\ell)$  average KL-error in estimating the conditional probabilities gets average error  $\delta_{KL}(\mathcal{A}_\ell) \leq I(\mathcal{M})/\ell + \hat{\delta}_{KL}(\mathcal{A}_\ell)$ .*

*Proof.* We will bound the expected error by splitting the time interval 0 to  $T - 1$  (during which predictions are made) into blocks of length  $\ell$ . Consider any block starting at time  $\tau$ . We find the

average error of the predictors from time  $\tau$  to  $\tau + \ell - 1$  and then average across all blocks.

To begin, note that we can decompose the error as the sum of the error in estimating  $\hat{P}$  and the error due to not knowing the past history. Consider any time  $t$ . Let  $Q_{\mathcal{A}_\ell}(x_t|x_{t-\ell+1}^{t-1}) = \hat{P}(x_t|x_{t-\ell+1}^{t-1})$  be the predictive distribution of the predictor  $\mathcal{A}_\ell$  at time  $t$ .

$$\begin{aligned}\delta_{KL}^{(t)}(\mathcal{A}_\ell) &= \mathbb{E}_{x_{-\infty}^{t-1}} \left[ D_{KL}(P(x_t|x_{-\infty}^{t-1}) \parallel \hat{P}(x_t|x_{t-\ell+1}^{t-1})) \right] \\ &= \mathbb{E}_{x_{-\infty}^{t-1}} \left[ D_{KL}(P(x_t|x_{-\infty}^{t-1}) \parallel P(x_t|x_{t-\ell+1}^{t-1})) \right] + \mathbb{E}_{x_{-\infty}^{t-1}} \left[ D_{KL}(P(x_t|x_{t-\ell+1}^{t-1}) \parallel \hat{P}(x_t|x_{t-\ell+1}^{t-1})) \right] \\ &= \delta_{KL}^{(t)}(\mathcal{P}_\ell) + \hat{\delta}_{KL}^{(t)}(\mathcal{A}_\ell)\end{aligned}$$

Therefore,  $\delta_{KL}(\mathcal{A}_\ell) = \delta_{KL}(\mathcal{P}_\ell) + \hat{\delta}_{KL}(\mathcal{A}_\ell)$ . It's easy to verify that  $\delta_{KL}^{(t)}(\mathcal{P}_\ell) = I(x_t; x_{-\infty}^{t-\ell} | x_{t-\ell+1}^{t-1})$ . Note that this relation expresses the intuition that the current output has a lot of extra information about the past if we cannot predict it as well as can be done by using the past. We will now upper bound the total error for the window  $[\tau, \tau + \ell - 1]$ . We expand  $I(x_{-\infty}^{\tau-1}; x_\tau^\infty)$  using the chain rule,

$$I(x_{-\infty}^{\tau-1}; x_\tau^\infty) = \sum_{t=\tau}^{\infty} I(x_{-\infty}^{\tau-1}; x_t | x_\tau^{t-1}) \geq \sum_{t=\tau}^{\tau+\ell-1} I(x_{-\infty}^{\tau-1}; x_t | x_\tau^{t-1}).$$

Note that  $I(x_{-\infty}^{\tau-1}; x_t | x_\tau^{t-1}) \geq I(x_{-\infty}^{t-\ell}; x_t | x_{t-\ell+1}^{t-1}) = \delta_{KL}^{(t)}(\mathcal{P}_\ell)$  as  $t - \ell \leq \tau - 1$  and  $I(X; Y; Z) \geq I(X; Z | Y)$ . The proposition now follows from averaging the error across the  $\ell$  time steps,

$$\frac{1}{\ell} \sum_{t=\tau}^{\tau+\ell-1} \delta_{KL}^{(t)}(\mathcal{P}_\ell) \leq \frac{1}{\ell} I(x_{-\infty}^{\tau-1}; x_\tau^\infty) = \frac{I(\mathcal{M})}{\ell}$$

□

The following easy corollary, relating KL error to  $\ell_1$  error yields the following statement, which also trivially applies to zero/one loss with respect to that of the optimal predictor, as the expected relative zero/one loss at any time step is at most the  $\ell_1$  loss at that time step.

**Corollary 2.** *For any data-generating distribution  $\mathcal{M}$  with mutual information  $I(\mathcal{M})$  between past and future observations, the best  $\ell$ -th order Markov model  $\mathcal{P}_\ell$  obtains average  $\ell_1$ -error,  $\delta_{\ell_1}(\mathcal{P}_\ell) \leq \sqrt{I(\mathcal{M})/2\ell}$  with respect to the optimal predictor with access to the infinite history. Also, any predictor  $\mathcal{A}_\ell$  with  $\hat{\delta}_{\ell_1}(\mathcal{A}_\ell)$  average  $\ell_1$ -error in estimating the conditional probabilities gets average error  $\delta_{\ell_1}(\mathcal{A}_\ell) \leq \sqrt{I(\mathcal{M})/2\ell} + \hat{\delta}_{\ell_1}(\mathcal{A}_\ell)$ .*

*Proof.* We again decompose the error as the sum of the error in estimating  $\hat{P}$  and the error due to not knowing the past history using the triangle inequality.

$$\begin{aligned}\delta_{\ell_1}^{(t)}(\mathcal{A}_\ell) &= \mathbb{E}_{x_{-\infty}^{t-1}} \left[ \| P(x_t|x_{-\infty}^{t-1}) - \hat{P}(x_t|x_{t-\ell+1}^{t-1}) \|_1 \right] \\ &\leq \mathbb{E}_{x_{-\infty}^{t-1}} \left[ \| P(x_t|x_{-\infty}^{t-1}) - P(x_t|x_{t-\ell+1}^{t-1}) \|_1 \right] + \mathbb{E}_{x_{-\infty}^{t-1}} \left[ \| P(x_t|x_{t-\ell+1}^{t-1}) - \hat{P}(x_t|x_{t-\ell+1}^{t-1}) \|_1 \right] \\ &= \delta_{\ell_1}^{(t)}(\mathcal{P}_\ell) + \hat{\delta}_{\ell_1}^{(t)}(\mathcal{A}_\ell)\end{aligned}$$

Therefore,  $\delta_{\ell_1}(\mathcal{A}_\ell) \leq \delta_{\ell_1}(\mathcal{P}_\ell) + \hat{\delta}_{\ell_1}(\mathcal{A}_\ell)$ . By Pinsker's inequality and Jensen's inequality,  $\delta_{\ell_1}^{(t)}(\mathcal{A}_\ell)^2 \leq \delta_{KL}^{(t)}(\mathcal{A}_\ell)/2$ . Using Proposition 1,

$$\delta_{KL}(\mathcal{A}_\ell) = \frac{1}{T} \sum_{t=0}^{T-1} \delta_{KL}^{(t)}(\mathcal{A}_\ell) \leq \frac{I(\mathcal{M})}{\ell}$$



Therefore, using Jensen’s inequality again,  $\delta_{\ell_1}(\mathcal{A}_\ell) \leq \sqrt{I(\mathcal{M})/2\ell}$ .  $\square$

Corollary 1 is an immediate consequence of Proposition 1. Consider any HMM  $\mathcal{M}$  with  $n$  hidden states and stationary distribution  $\pi$  with  $H(\pi)$  being the entropy of  $\pi$ . For HMMs, the hidden state of the HMM summarizes all the information in the past necessary to simulate the future, therefore  $I(\mathcal{M}) \leq H(\pi) \leq \log n$ .

### 3 Lower Bound for Large Alphabets

Our lower bounds for the sample complexity in the large alphabet case are based on the presumed hardness of distinguishing random instances of a certain CSP with high *complexity* from instances of that CSP with a high value. The complexity of a CSP is the largest  $r$  such that the CSP supports a  $(r - 1)$ -wise uniform distribution on its predicates but not a  $r$ -wise uniform distribution. The following conjecture on hardness of distinguishing planted CSP instances from random ones was made by Feldman et al. [10]. We define the notation more explicitly in the next subsection.

**Conjectured CSP Hardness [Conjecture 1]** [10]: *Let  $Q$  be any distribution over  $k$ -clauses and  $n$  variables of complexity  $r$  and  $0 < \eta < 1$ . Then any polynomial-time (randomized) algorithm that, given access to a distribution  $D$  that equals either the uniform distribution over  $k$ -clauses  $U_k$  or a (noisy) planted distribution  $Q_\sigma^\eta = (1 - \eta)Q_\sigma + \eta U_k$  for some  $\sigma \in \{0, 1\}^n$  and planted distribution  $Q_\sigma$ , decides correctly whether  $D = Q_\sigma^\eta$  or  $D = U_k$  with probability at least  $2/3$  needs  $\tilde{\Omega}(n^{r/2})$  clauses.*

Feldman et al. [10] proved the conjecture for the class of *statistical algorithms*<sup>3</sup>. Allen et al. [11] also suggested  $\tilde{\Omega}(n^{rk/2})$  to be the minimum number of clauses required to refute a random CSP with complexity  $r$ , and gave an algorithm to refute it beyond this regime. Mori and Witmer [12] made a similar conjecture (Conjecture 2 of their paper) which proposes that random CSPs cannot be refuted with fewer than  $\tilde{\Omega}(n^{rk/2})$  clauses with any polynomial sized SDP relaxation. ODonnell and Witmer [41] and Mori and Witmer [12] showed that the Sherali-Adams(SA)<sup>4</sup> SDP hierarchy cannot refute a random CSP below the  $\tilde{\Omega}(n^{rk/2})$  threshold. We also refer the reader to the work of Barak et al. [43] which shows that the stronger Sum-of-Squares hierarchy needs  $O(n)$  clauses when the predicate is pairwise uniform and previous work of Benabbas et al. [44] and Tulsiani and Worah [45] on hardness of refuting CSPs with a pairwise uniform predicate. As it seems difficult to base hardness of learning results directly on more standard assumptions like  $\mathbf{P} \neq \mathbf{NP}$ , other recent papers such as Daniely and Shalev-Shwartz [46] and Daniely [47] have also used presumed hardness of strongly refuting random  $k$ -SAT and random  $k$ -XOR instances with a small number of clauses to derive conditional hardness of learning results.

A first attempt to encode a  $k$ -CSP as a sequential model is to construct a model which outputs  $k$  randomly chosen literals for the first  $k$ -time steps, and then their (noisy) truth value for the final time step ( $k + 1$ ). Clauses from the CSP correspond to samples from the model, and the algorithm would need to solve the CSP to predict the final time step ( $k + 1$ ). However, as all the outputs up to the final time step are random and not predictable, the performance of any algorithm with respect to the optimal predictor will be good on average even if it does not do anything at the last

<sup>3</sup>Statistical algorithms are an extension of the statistical query model, these are algorithms that do not access samples from the distribution but instead have access to estimates of the expectation of any bounded function of a sample through an oracle. Feldman et al. [40] point out that almost all algorithms that work on random data also work with this limited access to samples, refer to Feldman et al. [40] for more details and examples.

<sup>4</sup>Polynomial sized SA relaxations are as powerful as any polynomial sized LP relaxation (see Chan et al. [42])

time step. Therefore, to get strong lower bound results, we need to do something extra and output  $m > 1$  functions of the  $k$ -literals after  $k$ -time steps, while still ensuring that all the functions remain collectively hard to invert without a large number of samples. We will use elementary results from the theory of error correcting codes to achieve this and prove hardness due to a reduction from a CSP which is hard based on Conjecture 1. By choosing  $k$  and  $m$  we can then get the desired dependence on the mutual information and error  $\epsilon$ . We provide a short outline of the argument, followed by the detailed proof.

*Outline of argument for lower bounds for large alphabet case:*

1. *Constructing a family of hard CSP instances  $\mathcal{C}_0$  using Conjecture 1:* For a fixed matrix  $\mathbf{A} \in \{0,1\}^{m \times k}$  for some  $k$  and  $m \leq k/2$ , the predicate  $P$  of the CSP  $\mathcal{C}_0$  is the set of all  $\mathbf{v} \in \{0,1\}^k$  such that  $\mathbf{A}\mathbf{v} = 0 \pmod 2$ . For any clause  $C$  and planted assignment  $\sigma$ , let  $\mathbf{v} = \sigma(C)$  where  $\sigma(C)$  is the  $k$ -bit string of values assigned by  $\sigma$  to literals in  $C$ . Hence for any planted assignment  $\sigma$ , the set of satisfying clauses of the CSP  $\mathcal{C}_0$  are all clauses such that  $\mathbf{v} = \sigma(C)$  is in the nullspace of  $\mathbf{A}$ . The distribution of clauses  $Q_{\sigma,0}^\eta$  is uniform on all satisfying clauses with probability  $(1 - \eta)$ , with probability  $\eta$  we add a uniformly random  $k$ -clause. Let  $U_k$  be the uniform distribution over all  $k$ -clauses. In subsection 3.2, we show how  $\mathbf{A}$  can be chosen to ensure that  $\mathcal{C}_0$  has complexity at least  $\gamma$ , where  $\gamma = 1/10$ . Then by Conjecture 1, any polynomial time algorithm cannot distinguish between the distribution  $Q_{\sigma,0}^\eta$  and  $U_k$  without sufficiently many clauses, or in other words any efficient algorithm cannot distinguish random instances of  $\mathcal{C}_0$  from instances of  $\mathcal{C}_0$  with a high value without sufficiently many clauses.
2. *Reducing instances in  $\mathcal{C}_0$  to instances from  $\mathcal{C}$ :* Consider a CSP  $\mathcal{C}$  defined by a collection of predicates  $P(\mathbf{y})$  for each  $\mathbf{y} \in \{0,1\}^m$ . For each  $\mathbf{y}$ , the predicate  $P(\mathbf{y})$  is the set of solutions to the system  $\mathbf{y} = \mathbf{A}\mathbf{v} \pmod 2$ . Hence each clause has an additional label  $\mathbf{y}$  which determines the satisfying assignments. For any clause  $C$ , we define  $\mathbf{v} = \sigma(C)$  as before. Hence for any planted assignment  $\sigma$ , the set of satisfying clauses of the CSP  $\mathcal{C}$  are all clauses such that  $\mathbf{y} = \mathbf{A}\mathbf{v} \pmod 2$  where  $\mathbf{y}$  is the label of the clause. The distribution of clauses  $Q_{\sigma}^\eta$  is uniform on all satisfying clauses with probability  $(1 - \eta)$ , with probability  $\eta$  we add a uniformly random  $k$ -clause. Lemma 1, shows that distinguishing  $Q_{\sigma}^\eta$  from  $U_k$  is hard if distinguishing between the distributions  $Q_{\sigma,0}^\eta$  and  $U_k$  is hard.
3. *Reducing instances in  $\mathcal{C}$  to instances from  $\mathcal{C}'$ :* We split the  $n$  unnegated variables into  $k$  sets, with the first  $n/k$  variables going the first set, the next  $n/k$  variables going into the next set and so on. Let the  $i$ th set of variables and their negations be  $\mathcal{R}_i$ . Let  $\mathcal{C}'$  be the CSP  $\mathcal{C}$  with the modification that the  $i$ th literal for each clause is chosen from the set  $\mathcal{R}_i$ . Lemma 3 shows that hardness results for distinguishing random instances of  $\mathcal{C}'$  from instances with a high value follows from hardness of distinguishing random instances of  $\mathcal{C}$  from instances of  $\mathcal{C}$  with a high value.
4. *Reducing instances in  $\mathcal{C}'$  to samples from sequential model  $\mathcal{M}$ :* We now show that we can construct a sequential model  $\mathcal{M}$  such that making good predictions on the model requires distinguishing random instances of  $\mathcal{C}'$  from instances with a high value. Intuitively, the model outputs  $k$  characters for the first  $k$  time steps, which correspond to literals in the CSP  $\mathcal{C}'$ . For the next  $m$  time steps, the model outputs  $\mathbf{y} \in \{0,1\}^m$  where  $\mathbf{y} = \mathbf{A}\mathbf{v} \pmod 2$ ,  $\mathbf{v} = \sigma(C)$  with  $C$  being the clause associated with the outputs of the first  $k$  time steps. Subsection 3.3.1 provides details of the model. Theorem

1 then proves conditional hardness results using the hardness of the family of instances  $\mathcal{C}'$ .

We provide a small example to illustrate the construction. Let  $k = 3$  and  $m = 1$ . Let  $\mathbf{A} \in \{0, 1\}^{1 \times 3}$ . The output alphabet of the model  $\mathcal{M}$  is  $\{a_i, 1 \leq i \leq 6\}$ . The letter  $a_1$  maps to the variable  $x_1$ ,  $a_2$  maps to  $\bar{x}_1$ , similarly  $a_3 \rightarrow x_2, a_4 \rightarrow \bar{x}_2, a_5 \rightarrow x_3, a_6 \rightarrow \bar{x}_3$ . Any assignment of the variables  $\{x_1, x_2, x_3\}$  determines a subset  $\mathcal{S}$  of  $\{a_i\}$  of size 3, where a letter is included in the subset if the corresponding variable is 1. Each choice of subset  $\mathcal{S}$  corresponds to a model  $\mathcal{M}$ , let  $\sigma$  be some planted assignment to  $\{x_1, x_2, x_3\}$  which defines a subset  $\mathcal{S}$  and hence a particular model  $\mathcal{M}$ . If the output of the model  $\mathcal{M}$  is  $a_1, a_3, a_6$  for the first three time steps, then this corresponds to the clause with literals,  $(x_1, x_2, \bar{x}_3)$ . For the final time step, with probability  $(1 - \eta)$  the model outputs  $y = \mathbf{A}\mathbf{v} \bmod 2$ , with  $\mathbf{v} = \sigma(C)$  for the clause  $C = (x_1, x_2, \bar{x}_3)$ , and with probability  $\eta$  it outputs a uniform random bit. For an algorithm to make a good prediction at the final time step, it needs to be able to distinguish if the output at the final time step is always a random bit or if it is dependent on the clause, hence it needs to distinguish random instances of the clause from noisy planted instances. This gives us the required conditional hardness results.

### 3.1 CSP formulation

We first go over some notation that we'll use for CSP problems, we follow the same notation and setup as in Feldman et al. [10]. Consider the following model for generating a random CSP instance on  $n$  variables with a satisfying assignment  $\sigma$ . The  $k$ -CSP is defined by the predicate  $P : \{0, 1\}^k \rightarrow \{0, 1\}$ . We represent a  $k$ -clause by an ordered  $k$ -tuple of literals from  $\{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$  with no repetition of variables and let  $X_k$  be the set of all such  $k$ -clauses. For a  $k$ -clause  $C = (l_1, \dots, l_k)$  let  $\sigma(C) \in \{0, 1\}^k$  be the  $k$ -bit string of values assigned by  $\sigma$  to literals in  $C$ , that is  $\{\sigma(l_1), \dots, \sigma(l_k)\}$  where  $\sigma(l_i)$  is the value of the literal  $l_i$  in assignment  $\sigma$ . In the planted model we draw clauses with probabilities that depend on the value of  $\sigma(C)$ . Let  $Q : \{0, 1\}^k \rightarrow \mathbb{R}^+$ ,  $\sum_{\mathbf{t} \in \{0, 1\}^k} Q(\mathbf{t}) = 1$  be some distribution over satisfying assignments to  $P$ . The distribution  $Q_\sigma$  is then defined as follows-

$$Q_\sigma(C) = \frac{Q(\sigma(C))}{\sum_{C' \in X_k} Q(\sigma(C'))} \quad (3.1)$$

Recall that for any distribution  $Q$  over satisfying assignments we define its complexity  $r$  as the largest  $r$  such that the distribution  $Q$  is  $(r - 1)$ -wise uniform (also referred to as  $(r - 1)$ -wise independent in the literature) but not  $r$ -wise uniform.

Consider the CSP  $\mathcal{C}$  defined by a collection of predicates  $P(\mathbf{y})$  for each  $\mathbf{y} \in \{0, 1\}^m$  for some  $m \leq k/2$ . Let  $\mathbf{A} \in \{0, 1\}^{m \times k}$  be a matrix with full row rank over the binary field. We will later choose  $\mathbf{A}$  to ensure the CSP has high complexity. For each  $\mathbf{y}$ , the predicate  $P(\mathbf{y})$  is the set of solutions to the system  $\mathbf{y} = \mathbf{A}\mathbf{v} \bmod 2$  where  $\mathbf{v} = \sigma(C)$ . For all  $\mathbf{y}$  we define  $Q_\mathbf{y}$  to be the uniform distribution over all consistent assignments, i.e. all  $\mathbf{v} \in \{0, 1\}^k$  satisfying  $\mathbf{y} = \mathbf{A}\mathbf{v} \bmod 2$ . The planted distribution  $Q_{\sigma, \mathbf{y}}$  is defined based on  $Q_\mathbf{y}$  according to equation 3.1. Each clause in  $\mathcal{C}$  is chosen by first picking a  $\mathbf{y}$  uniformly at random and then a clause from the distribution  $Q_{\sigma, \mathbf{y}}$ . For any planted  $\sigma$  we define  $Q_\sigma$  to be the distribution over all consistent clauses along with their labels

$\mathbf{y}$ . Let  $U_k$  be the uniform distribution over  $k$ -clauses, with each clause assigned a uniformly chosen label  $\mathbf{y}$ . Define  $Q_{\sigma}^{\eta} = (1 - \eta)Q_{\sigma} + \eta U_k$ , for some fixed noise level  $\eta > 0$ . We consider  $\eta$  to be a small constant less than 0.05. This corresponds to adding noise to the problem by mixing the planted and the uniform clauses. The problem gets harder as  $\eta$  becomes larger, for  $\eta = 0$  it can be efficiently solved using Gaussian Elimination. Note that the CSP  $\mathcal{C}$  is defined by a collection of predicates instead of a single one, whereas the setting for which hardness results have been conjectured corresponds to a single predicate for all clauses.

To get around this, we define another CSP  $\mathcal{C}_0$  which we show reduces to  $\mathcal{C}$ . The label  $\mathbf{y}$  is fixed to be the all zero vector in  $\mathcal{C}_0$ . Hence  $Q_0$ , the distribution over satisfying assignments for  $\mathcal{C}_0$ , is the uniform distribution over all vectors in the null space of  $\mathbf{A}$  over the binary field. We refer to the planted distribution in this case as  $Q_{\sigma,0}$ . Let  $U_{k,0}$  be the uniform distribution over  $k$ -clauses, with each clause now having the label 0. For any planted assignment  $\sigma$ , we denote the distribution of consistent clauses of  $\mathcal{C}_0$  by  $Q_{\sigma,0}$ . As before define  $Q_{\sigma,0}^{\eta} = (1 - \eta)Q_{\sigma,0} + \eta U_{k,0}$  for the same  $\eta$ .

Let  $L$  be the problem of distinguishing between  $U_k$  and  $Q_{\sigma}^{\eta}$  for some randomly and uniformly chosen  $\sigma \in \{0,1\}^n$  with success probability at least  $2/3$ . Similarly, let  $L_0$  be the problem of distinguishing between  $U_{k,0}$  and  $Q_{\sigma,0}^{\eta}$  for some randomly and uniformly chosen  $\sigma \in \{0,1\}^n$  with success probability at least  $2/3$ .  $L$  and  $L_0$  can be thought of as the problem of distinguishing random instances of the CSPs from instances with a high value. Note that  $L$  and  $L_0$  are at least as hard as the problem of refuting the random CSP instances  $U_k$  and  $U_{k,0}$ , as this corresponds to the case where  $\eta = 0$ . We claim that an algorithm for  $L$  implies an algorithm for  $L_0$ .

**Lemma 1.** *If  $L$  can be solved in time  $t(n)$  with  $s(n)$  clauses, then  $L_0$  can be solved in time  $O(t(n) + s(n))$  and  $s(n)$  clauses.*

Let the complexity of  $Q_0$  be  $\gamma k$ , with  $\gamma \geq 1/10$  (we demonstrate how to achieve this next). By Conjecture 1 distinguishing between  $U_{k,0}$  and  $Q_{\sigma,0}^{\eta}$  requires at least  $\tilde{\Omega}(n^{\gamma k/2})$  clauses. We now discuss how  $\mathbf{A}$  can be chosen to ensure that the complexity of  $Q_0$  is  $\gamma k$ .

### 3.2 Ensuring high complexity of the CSP

Let  $\mathcal{N}$  be the null space of  $\mathbf{A}$ . Note that the rank of  $\mathcal{N}$  is  $(k - m)$ . For any subspace  $\mathcal{D}$ , let  $\mathbf{w}(\mathcal{D}) = (w_1, w_2, \dots, w_k)$  be a randomly chosen vector from  $\mathcal{D}$ . To ensure that  $Q_0$  has complexity  $\gamma k$ , it suffices to show that the random variables  $\mathbf{w}(\mathcal{N}) = (w_1, w_2, \dots, w_k)$  are  $(\gamma k - 1)$ -wise uniform. We use the theory of error correcting codes to find such a matrix  $\mathbf{A}$ .

A binary linear code  $\mathcal{B}$  of length  $k$  and rank  $m$  is a linear subspace of  $\mathbb{F}_2^k$  (our notation is different from the standard notation in the coding theory literature to suit our setting). The generator matrix of the code is the matrix  $\mathbf{G}$  such that  $\mathcal{B} = \{\mathbf{G}\mathbf{v}, \mathbf{v} \in \{0,1\}^m\}$ . The parity check matrix of the code is the matrix  $\mathbf{H}$  such that  $\mathcal{B} = \{\mathbf{c} \in \{0,1\}^k : \mathbf{H}\mathbf{c} = 0\}$ . The distance  $d$  of a code is the weight of the minimum weight codeword. For any codeword  $\mathcal{B}$  we define its dual codeword  $\mathcal{B}^T$  as the codeword with generator matrix  $\mathbf{H}^T$  and parity check matrix  $\mathbf{G}^T$ . Note that the rank of the dual codeword of a code with rank  $m$  is  $(k - m)$ . We use the following standard result about linear codes-

**Fact 1.** *If  $\mathcal{B}^T$  has distance  $l$ , then  $\mathbf{w}(\mathcal{B})$  is  $(l - 1)$ -wise uniform.*

Hence, our job of finding  $\mathbf{A}$  reduces to finding a dual code with distance  $\gamma k$  and rank  $m$ . We use the Gilbert-Varshamov bound to argue for the existence of such a code. Let  $H(p)$  be the binary entropy of  $p$ , let  $\delta = d/k$  be the relative distance of any code.

**Lemma 2.** (*Gilbert-Varshamov bound*) For every  $0 \leq \delta < 1/2$ , and  $0 < \epsilon \leq 1 - H(\delta)$ , there exists a code with rank  $m$  and relative distance  $\delta$  if  $m/k \geq 1 - H(\delta) - \epsilon$ .

Taking  $\delta = 1/10$ , there exists a code  $\mathcal{B}$  whenever  $m/k \leq 0.5$ , which is the setting we're interested in. We choose  $\mathbf{A} = \mathbf{G}^T$ , where  $\mathbf{G}$  is the generator matrix of  $\mathcal{B}$ . Hence the null space of  $\mathbf{A}$  is  $(k/10 - 1)$ -wise uniform, hence the complexity of  $Q_0$  is  $\gamma k$  with  $\gamma \geq 1/10$ . Hence for all  $k$  and  $m \leq k/2$  we can find a  $\mathbf{A} \in \{0, 1\}^{m \times k}$  to ensure that the complexity of  $Q_0$  is  $\gamma k$ .

### 3.3 Sequential model of CSP and sample complexity lower bound

We now construct a sequential model which derives hardness from the hardness of  $L$ . We cannot base our model directly on  $L$  as generating random  $k$ -tuples without repetition increases the mutual information, so we formulate a slight variation  $L'$  of  $L$  which we show is at least as hard as  $L$ . We could not define our CSP instance allowing repetition as that is different from the setting examined in Feldman et al. [10], and hardness of the setting with repetition does not follow from hardness of the setting allowing repetition, though the converse is true.

#### 3.3.1 Constructing sequential model

Consider the following family of sequential models  $\mathcal{R}(n, \mathbf{A}_{m \times k})$  where  $\mathbf{A} \in \{0, 1\}^{m \times k}$  is chosen as defined previously. The output alphabet of all models in the family is  $\mathcal{X} = \{a_i, 1 \leq i \leq 2n\}$  of size  $2n$ , with  $2n/k$  even. We choose a subset  $\mathcal{S}$  of  $\mathcal{X}$  of size  $n$ , each choice of  $\mathcal{S}$  corresponds to a model  $\mathcal{M}$  in the family. Each letter in the output alphabet is encoded as a 1 or 0 which represents whether or not the letter is included in the set  $\mathcal{S}$ , let  $\mathbf{u} \in \{0, 1\}^{2n}$  be the vector which stores this encoding so  $u_i = 1$  whenever the letter  $a_i$  is in  $\mathcal{S}$ . Let  $\boldsymbol{\sigma} \in \{0, 1\}^n$  determine the subset  $\mathcal{S}$  such that entry  $u_{2i-1}$  is 1 and  $u_{2i}$  is 0 when  $\sigma_i$  is 1 and  $u_{2i-1}$  is 0 and  $u_{2i}$  is 1 when  $\sigma_i$  is 0, for all  $i$ . We choose  $\boldsymbol{\sigma}$  uniformly at random from  $\{0, 1\}^n$  and each choice of  $\boldsymbol{\sigma}$  represents some subset  $\mathcal{S}$ , and hence some model  $\mathcal{M}$ . We partition the output alphabet  $\mathcal{X}$  into  $k$  subsets of size  $2n/k$  each so the first  $2n/k$  letters go to the first subset, the next  $2n/k$  go to the next subset and so on. Let the  $i$ th subset be  $\mathcal{X}_i$ . Let  $\mathcal{S}_i$  be the set of elements in  $\mathcal{X}_i$  which belong to the set  $\mathcal{S}$ .

At time 0,  $\mathcal{M}$  chooses  $\mathbf{v} \in \{0, 1\}^k$  uniformly at random from  $\{0, 1\}^k$ . At time  $i, i \in \{0, \dots, k-1\}$ , if  $v_i = 1$ , then the model chooses a letter uniformly at random from the set  $\mathcal{S}_i$ , otherwise if  $v_i = 0$  it chooses a letter uniformly at random from  $\mathcal{X}_i - \mathcal{S}_i$ . With probability  $(1 - \eta)$  the outputs for the next  $m$  time steps from  $k$  to  $(k + m - 1)$  are  $\mathbf{y} = \mathbf{A}\mathbf{v} \bmod 2$ , with probability  $\eta$  they are  $m$  uniform random bits. The model resets at time  $(k + m - 1)$  and repeats the process.

We claim that  $I(\mathcal{M})$  is at most  $m$ . This is easy to verify because any information about the past gives us at most  $m$  bits of information about the future as we cannot hope to predict anything more about the future than the binary outputs from time  $k$  to  $(k + m - 1)$ . Also, it is not difficult to verify that  $\mathcal{M}$  can be simulated using a HMM.

#### 3.3.2 Reducing sequential model to CSP instance

We reveal the matrix  $\mathbf{A}$  to the algorithm, but the encoding  $\boldsymbol{\sigma}$  is kept secret. The task of finding the encoding  $\boldsymbol{\sigma}$  given samples from  $\mathcal{M}$  can be naturally seen as a CSP. Each sample is a clause with the literal corresponding to the output letter  $a_i$  being  $x_{(i+1)/2}$  whenever  $i$  is odd and  $\bar{x}_{i/2}$  when  $i$  is even. We refer the reader to the boxed outline at the beginning of the section for an example. We denote  $\mathcal{C}'$  as the CSP  $\mathcal{C}$  with the modification that the  $i$ th literal of each clause is the literal

corresponding to a letter in  $\mathcal{X}_i$  for all  $1 \leq i \leq k$ . Define  $Q'_\sigma$  as the distribution of consistent clauses for the CSP  $\mathcal{C}'$ . Define  $U'_k$  as the uniform distribution over  $k$ -clauses with the additional constraint that the  $i$ th literal of each clause is the literal corresponding to a letter in  $\mathcal{X}_i$  for all  $1 \leq i \leq k$ . Define  $Q'^\eta_\sigma = (1 - \eta)Q'_\sigma + \eta U'_k$ . Note that samples from the model  $\mathcal{M}$  are equivalent to clauses from  $Q'^\eta_\sigma$ . We now show that hardness of  $L'$  follows from hardness of  $L$ .

**Lemma 3.** *If  $L'$  can be solved in time  $t(n)$  with  $s(n)$  clauses, then  $L$  can be solved in time  $t(n)$  with  $O(s(n))$  clauses. Hence if Conjecture 1 is true then  $L'$  cannot be solved in polynomial time with less than  $\tilde{\Omega}(n^{\gamma^{k/2}})$  clauses.*

We can now prove the Theorem 1 using Lemma 3.

**Theorem 1.** *Assuming Conjecture 1, for all integers  $t > 0$  and  $0 < \epsilon \leq 0.1$ , there exists a family of distributions over sequences with observations drawn from an alphabet of size  $n$  such that every distribution  $\mathcal{M}$  in the family has mutual information of the past and future bounded as  $ct \leq I(\mathcal{M}) \leq t$  for some fixed constant  $c$ , yet any polynomial time algorithm that achieves average error KL-error,  $\ell_1$  error or relative zero-one error less than  $\epsilon$  with probability at least  $2/3$  over the choice of  $\mathcal{M}$  requires  $n^{\Theta(I(\mathcal{M})/\epsilon)}$  samples from  $\mathcal{M}$  over any window length which the algorithm uses for prediction.*

*Proof.* We describe how to choose the family of sequential models  $\mathcal{R}(n, \mathbf{A}_{m \times k})$  for each value of  $\epsilon$  and  $t$ . Given any  $\epsilon$  and  $t$  we choose  $m = t$  and  $k$  to be the solution of  $\frac{2}{9} \frac{m}{k+m} = \epsilon$ , rounded off to the previous largest integer. Note that  $\epsilon \leq 0.1$  therefore  $k \geq 2m$ . We choose the matrix  $\mathbf{A}_{m \times k}$  as outlined earlier. For each vector  $\sigma \in \{0, 1\}^n$  we define the family of sequential models  $\mathcal{R}(n, \mathbf{A})$  as earlier. Let  $\mathcal{M}$  be a randomly chosen model in the family. By our previous discussion,  $I(\mathcal{M}) \leq m$ .

We first show the result for the relative zero-one loss. The idea is that any algorithm which does a good job of predicting the outputs from time  $k$  through  $(k + m - 1)$  can be used to distinguish between instances of the CSP with a high value and uniformly random clauses. This is because it is not possible to make good predictions on uniformly random clauses. We relate the zero-one error from time  $k$  through  $(k + m - 1)$  with the relative zero-one error from time  $k$  through  $(k + m - 1)$  and the average zero-one error for all time steps to get the required lower bounds.

Let  $\rho_{01}(\mathcal{A})$  be the average zero-one loss of some polynomial time algorithm  $\mathcal{A}$  for the output time steps  $k$  through  $(k + m - 1)$  and  $\delta'_{01}(\mathcal{A})$  be the average relative zero-one loss of  $\mathcal{A}$  for the output time steps  $k$  through  $(k + m - 1)$  with respect to the optimal predictions. For the distribution  $U'_k$  it is not possible to get  $\rho_{01}(\mathcal{A}) < 0.5$  as the clauses and the label  $\mathbf{y}$  are independent and  $\mathbf{y}$  is chosen uniformly at random from  $\{0, 1\}^m$ . For  $Q'^\eta_\sigma$  it is information theoretically possible to get  $\rho_{01}(\mathcal{A}) = \eta$ . Hence any algorithm which gets error  $\rho_{01}(\mathcal{A}) \leq 2/5$  can be used to distinguish between  $U'_k$  and  $Q'^\eta_\sigma$ . Therefore by Lemma 3 any polynomial time algorithm which gets  $\rho_{01}(\mathcal{A}) \leq 2/5$  with probability greater than  $2/3$  over the choice of  $\mathcal{M}$  needs at least  $\tilde{\Omega}(n^{\gamma^{k/2}})$  samples. Note that  $\delta'_{01}(\mathcal{A}) = \rho_{01}(\mathcal{A}) - \eta$ . As the optimal predictor  $\mathcal{P}_\infty$  gets  $\rho_{01}(\mathcal{P}_\infty) = \eta < 0.05$ , therefore  $\delta'_{01}(\mathcal{A}) \leq 1/3 \implies \rho_{01}(\mathcal{A}) \leq 2/5$ . Note that  $\delta_{01}(\mathcal{A}) \geq \delta'_{01}(\mathcal{A}) \frac{m}{k+m}$ . This is because  $\delta_{01}(\mathcal{A})$  is the average error for all  $(k+m)$  time steps, and the contribution to the error from time steps  $0$  to  $(k-1)$  is non-negative. Also,  $\frac{1}{3} \frac{m}{k+m} > \epsilon$ , therefore,  $\delta_{01}(\mathcal{A}) < \epsilon \implies \delta'_{01}(\mathcal{A}) < \frac{1}{3} \implies \rho_{01}(\mathcal{A}) \leq 2/5$ . Hence any polynomial time algorithm which gets average relative zero-one loss less than  $\epsilon$  with probability greater than  $2/3$  needs at least  $\tilde{\Omega}(n^{\gamma^{k/2}})$  samples. The result for  $\ell_1$  loss follows directly from the result for relative zero-one loss, we next consider the KL loss.

Let  $\delta'_{KL}(\mathcal{A})$  be the average KL error of the algorithm  $\mathcal{A}$  from time steps  $k$  through  $(k+m-1)$ . By application of Jensen's inequality and Pinsker's inequality,  $\delta'_{KL}(\mathcal{A}) \leq 2/9 \implies \delta'_{01}(\mathcal{A}) \leq 1/3$ . Therefore, by our previous argument any algorithm which gets  $\delta'_{KL}(\mathcal{A}) < 2/9$  needs  $\tilde{\Omega}(n^{\gamma k/2})$  samples. But as before,  $\delta_{KL}(\mathcal{A}) \leq \epsilon \implies \delta'_{KL}(\mathcal{A}) \leq 2/9$ . Hence any polynomial time algorithm which succeeds with probability greater than  $2/3$  and gets average KL loss less than  $\epsilon$  needs at least  $\tilde{\Omega}(n^{\gamma k/2})$  samples.

We lower bound  $k$  by a linear function of  $I(\mathcal{M})/\epsilon$ . We claim that  $I(\mathcal{M})/\epsilon$  is at most  $10k$ . This follows because-

$$\begin{aligned} I(\mathcal{M})/\epsilon &\leq \frac{9m(k+m)}{2m} \\ &\leq 10k \end{aligned}$$

Hence any polynomial time algorithm needs  $n^{\Theta(I(\mathcal{M})/\epsilon)}$  samples to get average relative zero-one loss,  $\ell_1$  loss, or KL loss less than  $\epsilon$  on  $\mathcal{M}$ .

To finish, we verify that  $I(\mathcal{M})$  satisfies the required lower bound. We claim that  $(\gamma/20)t \leq I(\mathcal{M})$ . Note that  $I(\mathcal{M})/\epsilon$  is at least  $\gamma k/2$ . To verify note that by Proposition 1, the predictor  $\mathcal{P}_\ell$  gets KL error at most  $\epsilon$  for  $\ell = I(\mathcal{M})/\epsilon$ . As noted earlier,  $\delta_{KL}(\mathcal{P}_\ell) \leq \epsilon \implies \delta'_{KL}(\mathcal{P}_\ell) \leq 2/9 \implies \delta'_{01}(\mathcal{P}_\ell) \leq 1/3$ . Note that it is not information theoretically possible to get  $\delta_{01} < 1/3$  with window length  $\ell$  smaller than  $\gamma k/2$  as only looking at the past  $\gamma k/2$  outputs gives us no information about the next output. This follows because the distribution of clauses is  $\gamma k/2$ -wise uniform. Note that-

$$\begin{aligned} 9\epsilon/2 &\geq \frac{t}{k+t} \\ \implies \epsilon k &\geq t(2/9 - \epsilon) \\ \implies \epsilon k &\geq 0.1t \end{aligned}$$

Hence  $I(\mathcal{M})$  is at least  $(\gamma/20)t$ .

Hence for all integers  $t > 0$  and  $0 < \epsilon \leq 0.1$ , the model  $\mathcal{M}$  satisfying  $ct \leq I(\mathcal{M}) \leq t$  for a constant  $c$  exists and the Theorem follows.  $\square$

## 4 Lower Bound for Small Alphabets

Our lower bounds for the sample complexity in the binary case are based on the average case hardness of the decision version of the parity with noise problem. In the parity with noise problem on  $n$  bit inputs we are given examples  $\mathbf{v} \in \{0,1\}^n$  drawn uniformly from  $\{0,1\}^n$  along with their noisy labels  $\langle \mathbf{s}, \mathbf{v} \rangle + \epsilon \pmod 2$  where  $\mathbf{s} \in \{0,1\}^n$  is the (unknown) support of the parity function, and  $\epsilon \in \{0,1\}$  is the classification noise such that  $\mathbb{P}[\epsilon = 1] = \eta$  where  $\eta < 0.05$  is the noise level.

Let  $Q_{\mathbf{s}}^\eta$  be the distribution over examples of the parity with noise instance with  $\mathbf{s}$  as the support of the parity function and  $\eta$  as the noise level. Let  $U_n$  be the distribution over examples and labels where each labels is chosen uniformly from  $\{0,1\}$  independent of the example. The strength of our lower bounds depends on the level of hardness of parity with noise. Currently, the fastest algorithm for the problem due to Blum et al. [14] runs in time and samples  $2^{n/\log n}$ . We define the function  $f(n)$  as follows-

**Definition 1.**  $f(n)$  is the function such that for a uniformly random support  $\mathbf{s}$ , with probability at least  $(1 - 1/n^2)$  over the choice of  $\mathbf{s}$ , any (randomized) algorithm that can distinguish between  $Q_{\mathbf{s}}^\eta$  and  $U_n$  with success probability greater than  $2/3$  over the randomness of the examples and the algorithm needs  $f(n)$  time or samples.

We now define the natural sequential version of the problem. We denote the model as  $\mathcal{M}(\mathbf{A}_{m \times n})$  for some  $\mathbf{A} \in \{0, 1\}^{m \times n}$ ,  $m \leq n/2$ . From time 0 through  $(n - 1)$  the outputs of the model are i.i.d. and uniform on  $\{0, 1\}$ . Let  $\mathbf{v} \in \{0, 1\}^n$  be the vector of outputs from time 0 to  $(n - 1)$ . The outputs for the next  $m$  time steps are given by  $\mathbf{y} = \mathbf{A}\mathbf{v} + \epsilon \pmod 2$ , where  $\epsilon \in \{0, 1\}^m$  is the random noise and each entry  $\epsilon_i$  of  $\epsilon$  is an i.i.d random variable such that  $\mathbb{P}[\epsilon_i = 1] = \eta$ , where  $\eta$  is the noise level. Note that if  $\mathbf{A}$  is full row-rank, and  $\mathbf{v}$  is chosen uniformly at random from  $\{0, 1\}^n$ , the distribution of  $\mathbf{y}$  is uniform on  $\{0, 1\}^m$ .

We define a set of  $\mathbf{A}$  matrices, which will define a family of sequential models. Let  $\mathbf{A}'$  be the sub-matrix of  $\mathbf{A}$  corresponding to the first  $2n/3$  columns and all the  $m$  rows. Let  $\mathcal{S}$  be the set of all  $(m \times n)$  matrices  $\mathbf{A}$  such that the sub-matrix  $\mathbf{A}'$  is full row-rank. We denote  $\mathcal{R}$  as the family of models  $\mathcal{M}(\mathbf{A})$  for  $\mathbf{A} \in \mathcal{S}$ . Lemma 4 shows that with high probability over the choice of  $\mathbf{A}$ , distinguishing outputs from the model  $\mathcal{M}(\mathbf{A})$  from random examples  $U_n$  requires  $f(n)$  time or examples. The proof of Proposition 2 follows from Lemma 4 and is similar to the proof of Theorem 1.

**Lemma 4.** *Let  $\mathbf{A}$  be chosen uniformly at random from the set  $\mathcal{S}$ . Then, with probability at least  $(1 - 1/n)$  over the choice  $\mathbf{A} \in \mathcal{S}$ , any (randomized) algorithm that can distinguish the outputs from the model  $\mathcal{M}(\mathbf{A})$  from the distribution over random examples  $U_n$  with success probability greater than  $2/3$  over the randomness of the examples and the algorithm needs  $f(n)$  time or examples.*

**Proposition 2.** *With  $f(n)$  as defined in Definition 1, for all sufficiently large  $t$  and  $0 < \epsilon \leq 0.1$ , there exists a family of distributions over binary strings such that every distribution  $\mathcal{M}$  in the family has  $ct \leq I(\mathcal{M}) \leq t$  for some fixed constant  $c$ , and any algorithm that achieves average relative zero-one loss, average  $\ell_1$  loss, or average KL loss less than  $\epsilon$  with probability greater than  $2/3$  for a randomly chosen model in the family needs, requires  $f(I(\mathcal{M}/\epsilon))$  time or samples over any window length which the algorithm uses for prediction.*

## 5 Information theoretic lower bounds

We show that *information theoretically*, windows of length  $cI(\mathcal{M})/\epsilon^2$  are necessary to get expected relative zero-one loss less than  $\epsilon$ . As the expected relative zero-one loss is at most the  $\ell_1$  loss, which can be bounded by the square of the KL-divergence, this automatically implies that our window length requirement is also tight for  $\ell_1$  loss and KL loss. In fact, it's very easy to show the tightness for the KL loss, choose the simple model which emits uniform random bits from time 1 to  $n$  and repeats the bits from time 1 to  $m$  for the next  $m$  time steps. One can then choose  $n, m$  to get the desired error  $\epsilon$  and mutual information  $I(\mathcal{M})$ . To get a lower bound for the zero-one loss we use the probabilistic method to argue that there exists an HMM such that long windows are required to perform optimally with respect to the zero-one loss for that HMM.

**Proposition 3.** *There is an absolute constant  $c$  such that for all  $0 < \epsilon < 0.5$  and sufficiently large  $n$ , there exists an HMM with  $n$  states such that it is not information theoretically possible to get average relative zero-one loss or  $\ell_1$  loss less than  $\epsilon$  using windows of length smaller than  $c \log n / \epsilon^2$ , and KL loss less than  $\epsilon$  using windows of length smaller than  $c \log n / \epsilon$ .*



*Proof.* Consider a Hidden Markov Model with the Markov chain being a permutation on  $n$  states with  $n$  being a multiple of  $(\ell + 1)$ , where  $(\ell + 1) = c \log n / \epsilon^2$ , for a constant  $c = 1/33$ . We will regard  $\epsilon$  as a constant with respect to  $n$ . Let  $n/(\ell + 1) = t$ . We refer to the hidden states by  $h_i, 0 \leq i \leq (n - 1)$ ,  $h_i^j$  refers to the sequence of hidden states  $i$  through  $j$ . The output alphabet of each hidden state is binary. Each state  $i$  is marked with a label  $l_i$  which is 0 or 1, let  $G(i)$  be mapping from hidden state  $h_i$  to it's label  $l_i$ . All the states labeled 1 emit 1 with probability  $(0.5 + \epsilon)$  and 0 with probability  $(0.5 - \epsilon)$ . Similarly, all the states labeled 0 emit 0 with probability  $(0.5 + \epsilon)$  and 1 with probability  $(0.5 - \epsilon)$ .

We will show that a model looking at only the past  $\ell$  outputs cannot get average zero-one loss less than  $0.5 - o(1)$ . As the optimal prediction looking at all past outputs gets average zero-one loss  $0.5 - \epsilon + o(1)$  (as the hidden state at each time step can be determined to an arbitrarily high probability if we are allowed to look at an arbitrarily long past), this proves that windows of length  $\ell$  do not suffice to get average zero-one error with respect to the optimal predictions less than  $\epsilon - o(1)$ . Note that the Bayes optimal prediction at time  $(\ell + 1)$  to minimize the expected zero-one loss given outputs from time 1 to  $\ell$  is to predict the mode of the distribution  $P(x_{\ell+1}|x_1^\ell = s_1^\ell)$  where  $s_1^\ell$  is the sequence of outputs from time 1 to  $\ell$ . Also, note that  $P(x_{\ell+1}|x_1^\ell = s_1^\ell) = \sum_i P(h_{i_\ell=i}|x_1^\ell = s_1^\ell)P(x_{\ell+1}|h_{i_\ell=i})$  where  $h_{i_\ell}$  is the hidden state at time  $\ell$ . Hence the predictor is a weighted average of the prediction of each hidden state with the weight being the probability of being at that hidden state.

We index each state  $h_i$  of the permutation by a tuple  $(f(i), g(i)) = (j, k)$  where  $j = i \bmod (\ell + 1)$  and  $k = \lfloor \frac{i}{\ell+1} \rfloor$  hence  $0 \leq j \leq \ell$  and  $0 \leq k \leq (t - 1)$ . We help the predictor to make the prediction at time  $(\ell + 1)$  by providing it with the index  $f(i_\ell) = i_\ell \bmod (\ell + 1)$  of the true hidden state  $h_{i_\ell}$  at time  $\ell$ . The Bayes optimal prediction at time  $(\ell + 1)$  given outputs  $s_1^\ell$  from time 1 to  $\ell$  and index  $f(h_{i_\ell}) = j$  is to predict the mode of  $P(x_{\ell+1}|x_1^\ell = s_1^\ell, f(h_{i_\ell}) = j)$ . Note that by the definition of Bayes optimality, the average zero-one loss of the prediction using  $P(x_{\ell+1}|x_1^\ell = s_1^\ell, f(h_{i_\ell}) = j)$  cannot be worse than the average zero-one loss of the prediction using  $P(x_{\ell+1}|x_1^\ell = s_1^\ell)$ . We refer to the predictor using  $P(x_{\ell+1}|x_1^\ell = s_1^\ell, f(h_{i_\ell}) = j)$  as  $\mathcal{P}$ . We will now show that there exists some permutation for which the average zero-one loss of the predictor  $\mathcal{P}$  is  $0.5 - o(1)$ . We argue this using the probabilistic method. We choose a permutation uniformly at random from the set of all permutations. We show that the expected average zero-one loss of the predictor  $\mathcal{P}$  over the randomness in choosing the permutation is  $0.5 - o(1)$ . This means that there must exist some permutation such that the average zero-one loss of the predictor  $\mathcal{P}$  on that permutation is  $0.5 - o(1)$ .

To find the expected average zero-one loss of the predictor  $\mathcal{P}$  over the randomness in choosing the permutation, we will find the expected average zero-one loss of the predictor  $\mathcal{P}$  given that we are in some state  $h_{i_\ell}$  at time  $\ell$ . Without loss of generality let  $f(i_\ell) = 0$  and  $g(i_\ell) = (\ell - 1)$ , hence we were at the  $(\ell - 1)$ th hidden state at time  $\ell$ . Fix any sequence of labels for the hidden states  $h_0^{\ell-1}$ . For any string  $s_0^{\ell-1}$  emitted by the hidden states  $h_0^{\ell-1}$ , let  $\mathbb{E}[\delta(s_0^{\ell-1})]$  be the expected average zero-one error over the randomness in the rest of the permutation. Also, let  $\mathbb{E}[\delta(h_{\ell-1})] = \sum_{s_0^{\ell-1}} \mathbb{E}[\delta(s_0^{\ell-1})] \mathbb{P}[s_0^{\ell-1}]$  be the expected error averaged across all outputs. We will argue that  $\mathbb{E}[\delta(h_{\ell-1})] = 0.5 - o(1)$ . The set of hidden states  $h_i$  with  $g(i) = k$  defines a segment of the permutation, let  $r(k)$  be the label  $G(h_{(k-1)(\ell+1)}^{k(\ell+1)-2})$  of the segment  $k$ , excluding it's last bit which corresponds to the predictions. Let  $\mathcal{S}_1 = \{r(k), \forall k \neq 0\}$  be the set of all the labels excluding the first label  $r(0)$  and  $\mathcal{S}_2 = \{G(h_{k(\ell+1)+\ell}), \forall k\}$  be the set of all the predicted bits (refer to the figure for an example). Consider any assignment of  $r(0)$ . To begin, we show that with high probability

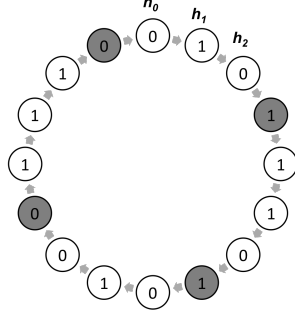


Figure 1: Lower bound construction,  $\ell = 3, n = 16$ . Each state is assigned a label which determines its output distribution. The state labeled 0 emit 0 with probability  $0.5 + \epsilon$  and the states labeled 1 emit 1 with probability  $0.5 + \epsilon$ . We consider we are at state  $h_{i_\ell} = h_2$  at time 2.  $r(0)$  corresponds to the label of  $h_0, h_1$  and  $h_2$  and is  $(0, 1, 0)$  in this case. Similarly,  $r(1) = (1, 1, 0)$  in this case. The segments between the shaded nodes comprise the set  $\mathcal{S}_1$  and are the possible sequences of states from which the last  $\ell = 3$  outputs could have come. The shaded nodes correspond to the states in  $\mathcal{S}_2$ , and are the possible predictions for the next time step. In this example  $\mathcal{S}_1 = \{(0, 1, 0), (1, 1, 0), (0, 1, 0), (1, 1, 1)\}$  and  $\mathcal{S}_2 = \{1, 1, 0, 0\}$ . The proof idea is that with high probability the output  $s_0^\ell$  from the segment  $r(0)$  corresponding to the hidden states  $(h_0, h_1, h_2)$  is close in Hamming distance to the label of some other segment, say  $r(2) = (0, 1, 0)$  and as the output distribution for the next state (the shaded nodes) is independent for the segments  $r(0)$  and  $r(2)$ , and hence any predictor cannot make a good prediction.

over the output  $s_0^{\ell-1}$ , the Hamming distance  $D(s_0^{\ell-1}, r(0))$  of the output  $s_0^{\ell-1}$  of the set of hidden states  $h_0^{\ell-1}$  from  $r(0)$  is at least  $\frac{\ell}{2} - 2\epsilon\ell$ . This follows directly from Hoeffding's inequality as all the outputs are independent conditioned on the hidden state-

$$\mathbb{P}[D(s_0^{\ell-1}, r(0)) \leq \ell/2 - 2\epsilon\ell] \leq e^{-2\ell\epsilon^2} \leq n^{-2c} \quad (5.1)$$

We now show that for any  $k \neq 0$  with decent probability the label  $r(k)$  of the segment  $k$  is closer in Hamming distance to the output  $s_0^{\ell-1}$  than  $r(0)$ . Then we argue that with high probability there are many such segments which are closer to  $s_0^{\ell-1}$  in Hamming distance than  $r(0)$ . Hence these other segments are assigned as much weight in predicting the next output as  $r(0)$ , which means that the output cannot be predicted with a high accuracy as the output bits corresponding to different segments are independent.

We first find the probability that the segment corresponding to some  $k$  with label  $r(k)$  has a Hamming distance less than  $\frac{\ell}{2} - \sqrt{\ell \log t/8}$  from any fixed binary string  $x$  of length  $\ell$ . Let  $F(l, m, p)$  be the probability of getting at least  $l$  heads in  $m$  i.i.d. trials with each trial having probability  $p$  of giving a head.  $F(l, n, p)$  can be bounded below by the following standard inequality-

$$F(l, m, p) \geq \frac{1}{\sqrt{2m}} \exp\left(-mD_{KL}\left(\frac{l}{m} \parallel p\right)\right)$$

We can use this to lower bound  $\mathbb{P}[\delta(r(k), x) \leq \ell/2 - \sqrt{\ell \log t/8}]$ ,

$$\begin{aligned} \mathbb{P}[D(r(k), x) \leq \ell/2 - \sqrt{\ell \log t/8}] &= F(\ell/2 + \sqrt{\ell \log t/8}, \ell, 1/2) \\ &\geq \frac{1}{\sqrt{2\ell}} \exp\left(-\ell D_{KL}\left(\frac{1}{2} + \sqrt{\frac{\log t}{8\ell}} \parallel \frac{1}{2}\right)\right) \end{aligned}$$

Note that  $D_{KL}(\frac{1}{2} + v \parallel v) \leq 4v^2$  by using the inequality  $\log(1 + v) \leq v$ . We can simplify the KL-divergence using this and write-

$$\mathbb{P}\left[D(r(k), x) \leq \ell/2 - \sqrt{\ell \log t/8}\right] \geq 1/\sqrt{2\ell t} \quad (5.2)$$

Let  $\mathcal{D}$  be the set of all  $k \neq 0$  such that  $D(r(k), x) \leq \ell/2 - \sqrt{\ell \log t/8}$  for some fixed  $x$ . We argue that with high probability over the randomness of the permutation  $|\mathcal{D}|$  is large. This follows from Eq. 5.2 and the Chernoff bound as the labels for all segments  $r(k)$  are chosen independently-

$$\mathbb{P}\left[|\mathcal{D}| \leq \sqrt{t/(8\ell)}\right] \leq e^{-\frac{1}{8}\sqrt{t/(2\ell)}}$$

Note that  $\sqrt{t/(8\ell)} \geq n^{0.25}$ . Therefore for any fixed  $x$ , with probability  $1 - \exp(-\frac{1}{8}\sqrt{\frac{t}{2\ell}}) \geq 1 - n^{-0.25}$  there are  $\sqrt{\frac{t}{8\ell}} \geq n^{0.25}$  segments in a randomly chosen permutation which have Hamming distance less than  $\ell/2 - \sqrt{\ell \log t/8}$  from  $x$ . By our construction  $2\epsilon\ell \leq \sqrt{\ell \log t/8}$  because  $\log(\ell + 1) \leq (1 - 32c) \log n$ .

Therefore if  $D(s_0^{\ell-1}, r(0)) > \ell/2 - 2\epsilon\ell$ , then with high probability over randomly choosing the segments  $\mathcal{S}_1$  there is a subset  $\mathcal{D}$  of segments in  $\mathcal{S}_1$  with  $|\mathcal{D}| \geq n^{0.25}$  such that all of the segments in  $\mathcal{D}$  have Hamming distance less than  $D(s_0^{\ell-1}, r(0))$  from  $s_0^{\ell-1}$ . Pick any  $s_0^{\ell-1}$  such that  $D(s_0^{\ell-1}, r(0)) > \ell/2 - 2\epsilon\ell$ . Consider any set of segments  $\mathcal{S}_1$  which has such a subset  $\mathcal{D}$  with respect to the string  $s_0^{\ell-1}$ . For all such permutations, the predictor  $\mathcal{P}$  places at least as much weight on the hidden states  $h_i$  with  $g(i) = k$ , with  $k$  such that  $r(k) \in \mathcal{D}$  as the true hidden state  $h_{\ell-1}$ . The prediction for any hidden state  $h_i$  is the corresponding bit in  $\mathcal{S}_2$ . Notice that the bits in  $\mathcal{S}_2$  are independent and uniform as we've not used them in any argument so far. The average correlation of an equally weighted average of  $m$  independent and uniform random bits with any one of the random bits is at most  $1/\sqrt{m}$ . Hence over the randomness of  $\mathcal{S}_2$ , the expected zero-one loss of the predictor is at least  $0.5 - n^{-0.1}$ . Hence we can write-

$$\begin{aligned} \mathbb{E}[\delta(s_0^{\ell-1})] &\geq (0.5 - n^{-0.1})\mathbb{P}[|\mathcal{D}| \geq \sqrt{t/(8\ell)}] \\ &\geq (0.5 - n^{-0.1})(1 - e^{-n^{0.25}}) \\ &\geq 0.5 - 2n^{-0.1} \end{aligned}$$

By using Equation 5.1, for any assignment  $r(0)$  to  $h_0^{\ell-1}$

$$\begin{aligned} \mathbb{E}[\delta(h_{\ell-1})] &\geq \mathbb{P}\left[D(s_0^{\ell-1}, r(0)) > \ell/2 - 2\epsilon\ell\right] \mathbb{E}\left[\delta(s_0^{\ell-1}) \mid D(s_0^{\ell-1}, r(0)) > \ell/2 - 2\epsilon\ell\right] \\ &\geq (1 - n^{-2c})(0.5 - 2n^{-0.1}) \\ &= 0.5 - o(1) \end{aligned}$$

As this is true for all assignments  $r(0)$  to  $h_0^{\ell-1}$  and for all choices of hidden states at time  $\ell$ , using linearity of expectations and averaging over all hidden states, the expected average zero-one loss of the predictor  $\mathcal{P}$  over the randomness in choosing the permutation is  $0.5 - o(1)$ . This means that there must exist some permutation such that the average zero-one loss of the predictor  $\mathcal{P}$  on that permutation is  $0.5 - o(1)$ . Hence there exists an HMM on  $n$  states such that is not information theoretically possible to get average zero-one error with respect to the optimal predictions less than  $\epsilon - o(1)$  using windows of length smaller than  $c \log n / \epsilon^2$  for a fixed constant  $c$ .

Therefore, for all  $0 < \epsilon < 0.5$  and sufficiently large  $n$ , there exists an HMM with  $n$  states such that it is not information theoretically possible to get average relative zero-one loss less than  $\epsilon/2 < \epsilon - o(1)$  using windows of length smaller than  $c\epsilon^{-2} \log n$ . The result for relative zero-one loss follows on replacing  $\epsilon/2$  by  $\epsilon'$  and setting  $c' = c/4$ . The result follows immediately from this as the expected relative zero-one loss is less than the expected  $\ell_1$  loss. For KL-loss we use Pinsker's inequality and Jensen's inequality.  $\square$

## A Additional proofs from Section 4

### A.1 Proof of Lemma 1

**Lemma 1.** *If  $L$  can be solved in time  $t(n)$  with  $s(n)$  clauses, then  $L_0$  can be solved in time  $O(t(n) + s(n))$  and  $s(n)$  clauses.*

*Proof.* We show that a random instance of  $\mathcal{C}_0$  can be transformed to a random instance of  $\mathcal{C}$  in time  $s(n)O(k)$  by independently transforming every clause  $C$  in  $\mathcal{C}_0$  to a clause  $C'$  in  $\mathcal{C}$  such that  $C$  is satisfied in the original CSP  $\mathcal{C}_0$  with some assignment  $\mathbf{t}$  to  $\mathbf{x}$  if and only if the corresponding clause  $C'$  in  $\mathcal{C}$  is satisfied with the same assignment  $\mathbf{t}$  to  $\mathbf{x}$ . For every  $\mathbf{y} \in \{0, 1\}^m$  we pre-compute and store a random solution of the system  $\mathbf{y} = \mathbf{A}\mathbf{v} \pmod 2$ , let the solution be  $v(\mathbf{y})$ . Given any clause  $C = (x_1, x_2, \dots, x_k)$  in  $\mathcal{C}_0$ , choose  $\mathbf{y} \in \{0, 1\}^m$  uniformly at random. We generate a clause  $C' = (x'_1, x'_2, \dots, x'_k)$  in  $\mathcal{C}$  from the clause  $C$  in  $\mathcal{C}_0$  by choosing the literal  $x'_i = \bar{x}_i$  if  $v_i(\mathbf{y}) = 1$  and  $x'_i = x_i$  if  $v_i(\mathbf{y}) = 0$ . By the linearity of the system, the clause  $C'$  is a consistent clause of  $\mathcal{C}$  with some assignment  $\mathbf{x} = \mathbf{t}$  if and only if the clause  $C$  was a consistent clause of  $\mathcal{C}_0$  with the same assignment  $\mathbf{x} = \mathbf{t}$ .

We next claim that  $C'$  is a randomly generated clause from the distribution  $U_k$  if  $C$  was drawn from  $U_{k,0}$  and is a randomly generated clause from the distribution  $Q_\sigma$  if  $C$  was drawn from  $Q_{\sigma,0}$ . By our construction, the label of the clause  $\mathbf{y}$  is chosen uniformly at random. Note that choosing a clause uniformly at random from  $U_{k,0}$  is equivalent to first uniformly choosing a  $k$ -tuple of unnegated literals and then choosing a negation pattern uniformly at random. It is clear that a clause is still uniformly random after adding another negation pattern if it was uniformly random before. Hence, if the original clause  $C$  was drawn to the uniform distribution  $U_{k,0}$ , then  $C'$  is distributed according to  $U_k$ . Similarly, choosing a clause uniformly at random from  $Q_{\sigma,y}$  for some  $\mathbf{y}$  is equivalent to first uniformly choosing a  $k$ -tuple of unnegated literals and then choosing a negation pattern uniformly at random which makes the clause consistent. As the original negation pattern corresponds to a  $\mathbf{v}$  randomly chosen from the null space of  $\mathbf{A}$ , the final negation pattern on adding  $\mathbf{v}(\mathbf{y})$  corresponds to the negation pattern for a uniformly random chosen solution of  $\mathbf{y} = \mathbf{A}\mathbf{v} \pmod 2$  for the chosen  $\mathbf{y}$ . Therefore, the clause  $C'$  is a uniformly random chosen clause from  $Q_{\sigma,y}$  if  $C$  is a uniformly random chosen clause from  $Q_{\sigma,0}$ .

Hence if it is possible to distinguish  $U_k$  and  $Q_\sigma^\eta$  for some randomly chosen  $\sigma \in \{0, 1\}^n$  with success probability at least  $2/3$  in time  $t(n)$  with  $s(n)$  clauses, then it is possible to distinguish between  $U_{k,0}$  and  $Q_{\sigma,0}^\eta$  for some randomly chosen  $\sigma \in \{0, 1\}^n$  with success probability at least  $2/3$  in time  $t(n) + s(n)O(k)$  with  $s(n)$  clauses.  $\square$

### A.2 Proof of Lemma 3

**Lemma 3.** *If  $L'$  can be solved in time  $t(n)$  with  $s(n)$  clauses, then  $L$  can be solved in time  $t(n)$  with  $O(s(n))$  clauses. Hence if Conjecture 1 is true then  $L'$  cannot be solved in polynomial time*

with less than  $\tilde{\Omega}(n^{\gamma^{k/2}})$  clauses.

*Proof.* Define  $E$  to be the event that a clause generated from the distribution  $Q_{\sigma}$  of the CSP  $\mathcal{C}$  has the property that for all  $i$  the  $i$ th literal belongs to the set  $\mathcal{X}_i$ , we also refer to this property of the clause as  $E$  for notational ease. It's easy to verify that the probability of the event  $E$  is  $1/k^k$ . We claim that conditioned on the event  $E$ , the CSP  $\mathcal{C}$  and  $\mathcal{C}'$  are equivalent.

This is verified as follows. Note that for all  $\mathbf{y}$ ,  $Q_{\sigma, \mathbf{y}}$  and  $Q'_{\sigma, \mathbf{y}}$  are uniform on all consistent clauses. Let  $\mathcal{U}$  be the set of all clauses with non-zero probability under  $Q_{\sigma, \mathbf{y}}$  and  $\mathcal{U}'$  be the set of all clauses with non-zero probability under  $Q'_{\sigma, \mathbf{y}}$ . Furthermore, for any  $\mathbf{v}$  which satisfies the constraint that  $\mathbf{y} = \mathbf{A}\mathbf{v} \pmod 2$ , let  $\mathcal{U}(\mathbf{v})$  be the set of clauses  $C \in \mathcal{U}$  such that  $\sigma(C) = \mathbf{v}$ . Similarly, let  $\mathcal{U}'(\mathbf{v})$  be the set of clauses  $C \in \mathcal{U}'$  such that  $\sigma(C) = \mathbf{v}$ . Note that the subset of clauses in  $\mathcal{U}(\mathbf{v})$  which satisfy  $E$  is the same as the set  $\mathcal{U}'(\mathbf{v})$ . As this holds for every consistent  $\mathbf{v}$  and the distributions  $Q'_{\sigma, \mathbf{y}}$  and  $Q_{\sigma, \mathbf{y}}$  are uniform on all consistent clauses, the distribution of clauses from  $Q_{\sigma}$  is identical to the distribution of clauses  $Q'_{\sigma}$  conditioned on the event  $E$ . The equivalence of  $U_k$  and  $U'_k$  conditioned on  $E$  also follows from the same argument.

Note that as the  $k$ -tuples in  $\mathcal{C}$  are chosen uniformly at random from satisfying  $k$ -tuples, with high probability there are  $s(n)$  tuples having property  $E$  if there are  $O(k^k s(n))$  clauses in  $\mathcal{C}$ . As the problems  $L$  and  $L'$  are equivalent conditioned on event  $E$ , if  $L'$  can be solved in time  $t(n)$  with  $s(n)$  clauses, then  $L$  can be solved in time  $t(n)$  with  $O(k^k s(n))$  clauses. From Lemma 1 and Conjecture 1,  $L$  cannot be solved in polynomial time with less than  $\tilde{\Omega}(n^{\gamma^{k/2}})$  clauses. Hence  $L'$  cannot be solved in polynomial time with less than  $\tilde{\Omega}(n^{\gamma^{k/2}}/k^k)$  clauses. As  $k$  is a constant with respect to  $n$ ,  $L'$  cannot be solved in polynomial time with less than  $\tilde{\Omega}(n^{\gamma^{k/2}})$  clauses.  $\square$

## B Additional proofs from Section 4

### B.1 Proof of Lemma 4

**Lemma 4.** *Let  $\mathbf{A}$  be chosen uniformly at random from the set  $\mathcal{S}$ . Then, with probability at least  $(1 - 1/n)$  over the choice  $\mathbf{A} \in \mathcal{S}$ , any (randomized) algorithm that can distinguish the outputs from the model  $\mathcal{M}(\mathbf{A})$  from the distribution over random examples  $U_n$  with success probability greater than  $2/3$  over the randomness of the examples and the algorithm needs  $f(n)$  time or examples.*

*Proof.* Suppose  $\mathbf{A} \in \{0, 1\}^{m \times n}$  is chosen at random with each entry being i.i.d. with its distribution uniform on  $\{0, 1\}$ . We claim that  $P(\mathbf{A} \in \mathcal{S}) \geq 1 - m2^{-n/6}$ . To verify, consider the addition of each row one by one to  $\mathbf{A}'$ . The probability of the  $i$ th row being linearly dependent on the previous  $(i - 1)$  rows is  $2^{i-1-2n/3}$ . Hence by a union bound,  $\mathbf{A}'$  is full row-rank with failure probability at most  $m2^{m-2n/3} \leq m2^{-n/6}$ . From Definition 1 and a union bound, any algorithm that can distinguish the outputs from the model  $\mathcal{M}(\mathbf{A})$  for uniformly chosen  $\mathbf{A}$  from the distribution over random examples  $U_n$  with probability at least  $(1 - 1/(2n))$  over the choice of  $\mathbf{A}$  needs  $f(n)$  time or examples. As  $P(\mathbf{A} \in \mathcal{A}) \geq 1 - m2^{-n/6}$ , for a uniformly chosen  $\mathbf{A} \in \mathcal{S}$ , with probability at least  $(1 - 1/(2n) - m2^{-n/6}) \geq (1 - 1/n)$  over the choice  $\mathbf{A} \in \mathcal{S}$  any algorithm that can distinguish the outputs from the model  $\mathcal{M}(\mathbf{A})$  from the distribution over random examples  $U_n$  with success probability greater than  $2/3$  over the randomness of the examples and the algorithm needs  $f(n)$  time or examples.  $\square$

## B.2 Proof of Proposition 2

**Proposition 2.** *With  $f(n)$  as defined in Definition 1, for all sufficiently large  $t$  and  $0 < \epsilon \leq 0.1$ , there exists a family of distributions over binary strings such that every distribution  $\mathcal{M}$  in the family has  $ct \leq I(\mathcal{M}) \leq t$  for some fixed constant  $c$ , and any algorithm that achieves average relative zero-one loss, average  $\ell_1$  loss, or average KL loss less than  $\epsilon$  with probability greater than  $2/3$  for a randomly chosen model in the family needs, requires  $f(I(\mathcal{M})/\epsilon)$  time or samples from  $\mathcal{M}$  over any window length which the algorithm uses for prediction.*

*Proof.* We describe the family for each value of  $\epsilon$  and  $t$ . Given any  $\epsilon$  and  $t$  we choose  $n$  to be the solution of  $\frac{2}{9} \frac{t}{n+t} = \epsilon$ , rounded off to the previous largest integer and  $m = t$ . Note that  $\epsilon < 1/2$  therefore  $n \geq 2m$ . The family is defined by the model  $\mathcal{M}(\mathbf{A}_{m \times n})$  defined previously with the matrix  $\mathbf{A}_{m \times n}$  chosen uniformly at random from the set  $\mathcal{S}$ . Note that  $I[\mathcal{M}(\mathbf{A})]$  is at most  $m$ . This is easy to verify because any information about the past gives us at most  $m$  bits of information about the future as we cannot hope to predict anything more about the future than the binary outputs from time  $n$  to  $(n + m - 1)$ .

Let  $\rho_{01}(\mathcal{A})$  be the average zero-one loss of some algorithm  $\mathcal{A}$  for the output time steps  $n$  through  $(n + m - 1)$  and  $\delta'_{01}(\mathcal{A})$  be the average relative zero-one loss of  $\mathcal{A}$  for the output time steps  $n$  through  $(n + m - 1)$  with respect to the optimal predictions. For the distribution  $U_n$  it is not possible to get  $\rho_{01}(\mathcal{A}) < 0.5$  as the clauses and the label  $\mathbf{y}$  are independent and  $\mathbf{y}$  is chosen uniformly at random from  $\{0, 1\}^m$ . For  $Q_s^\eta$  it is information theoretically possible to get  $\rho_{01}(\mathcal{A}) = \eta$ . Hence any algorithm which gets error  $\rho_{01}(\mathcal{A}) \leq 2/5$  can be used to distinguish between  $U_n$  and  $Q_s^\eta$ . Therefore by Lemma 4 any algorithm which gets  $\rho_{01}(\mathcal{A}) \leq 2/5$  with probability greater than  $2/3$  over the choice of  $\mathcal{M}(\mathbf{A})$  needs at least  $f(n)$  time or samples. Note that  $\delta'_{01}(\mathcal{A}) = \rho_{01}(\mathcal{A}) - \eta$ . This is because  $\delta_{01}(\mathcal{A})$  is the average error for all  $(n + m)$  time steps, and the contribution to the error from time steps 0 to  $(n - 1)$  is non-negative. As the optimal predictor  $\mathcal{P}_\infty$  gets  $\rho_{01}(\mathcal{P}_\infty) = \eta < 0.05$ , therefore  $\delta'_{01}(\mathcal{A}) \leq 1/3 \implies \rho_{01}(\mathcal{A}) \leq 2/5$ . Note that  $\delta_{01}(\mathcal{A}) \geq \delta'_{01}(\mathcal{A}) \frac{m}{n+m}$ . Also,  $\frac{1}{3} \frac{m}{n+m} > \epsilon$ , therefore,  $\delta_{01}(\mathcal{A}) < \epsilon \implies \delta'_{01}(\mathcal{A}) < \frac{1}{3} \implies \rho_{01}(\mathcal{A}) \leq 2/5$ . Hence any algorithm which gets average relative zero-one loss less than  $\epsilon$  with probability greater than  $2/3$  over the choice of  $\mathcal{M}(\mathbf{A})$  needs  $f(n)$  time or samples. The result for  $\ell_1$  loss follows directly from the result for relative zero-one loss, we next consider the KL loss.

Let  $\delta'_{KL}(\mathcal{A})$  be the average KL error of the algorithm  $\mathcal{A}$  from time steps  $n$  through  $(n + m - 1)$ . By application of Jensen's inequality and Pinsker's inequality,  $\delta'_{KL}(\mathcal{A}) \leq 2/9 \implies \delta'_{01}(\mathcal{A}) \leq 1/3$ . Therefore, by our previous argument any algorithm which gets  $\delta'_{KL}(\mathcal{A}) < 2/9$  needs  $f(n)$  samples. But as before,  $\delta_{KL}(\mathcal{A}) \leq \epsilon \implies \delta'_{KL}(\mathcal{A}) \leq 2/9$ . Hence any algorithm which gets average KL loss less than  $\epsilon$  needs  $f(n)$  time or samples.

We lower bound  $n$  by a linear function of  $I(\mathcal{M}(\mathbf{A}))/\epsilon$ . We claim that  $I(\mathcal{M}(\mathbf{A}))/\epsilon$  is at most  $10k$ . This follows because-

$$\begin{aligned} I(\mathcal{M}(\mathbf{A}))/\epsilon &\leq \frac{9m(n+m)}{2m} \\ &\leq 10n \end{aligned}$$

Hence any algorithm needs  $f(I(\mathcal{M}(\mathbf{A}))/\epsilon)$  samples and time to get average relative zero-one loss,  $\ell_1$  loss, or KL loss less than  $\epsilon$  with probability greater than  $2/3$  over the choice of  $\mathcal{M}(\mathbf{A})$ .

To finish, we verify that  $I(\mathcal{M}(\mathbf{A}))$  satisfies the required lower bound. We claim that  $I(\mathcal{M}(\mathbf{A})) \geq t/20$ . Let  $\ell = I(\mathcal{M}(\mathbf{A}))/\epsilon$ . By Proposition 1, the predictor  $\mathcal{P}_\ell$  gets KL error at most  $\epsilon$ . As noted earlier,  $\delta_{KL}(\mathcal{P}_\ell) \leq \epsilon \implies \delta'_{KL}(\mathcal{P}_\ell) \leq 2/9 \implies \delta'_{01}(\mathcal{P}_\ell) \leq 1/3$ . Note that it is not information theoretically possible to get  $\delta'_{01} < 1/3$  with windows smaller than  $n/3$  as only looking at the past  $n/3$  outputs gives us no information about the next output. This is because the matrix  $\mathbf{A}'$  is full row-rank, hence the dependence of the parity bits on the inputs from time 1 to  $2n/3$  cannot be inferred from the outputs after time  $2n/3$ . This is the reason that we required the submatrix  $\mathbf{A}'$  of  $\mathbf{A}$  to be full row rank. Therefore  $\ell \geq n/3$ . We now lower bound  $\epsilon n$  as follows-

$$\begin{aligned} 9\epsilon/2 &\geq \frac{t}{n+t} \\ \implies \epsilon n &\geq t(2/9 - \epsilon) \\ \implies \epsilon n &\geq 0.1t \end{aligned}$$

Hence  $I(\mathcal{M}(\mathbf{A}))$  is at least  $n\epsilon/3 \geq t/30$ . Hence the model  $\mathcal{M}(\mathbf{A})$  satisfies  $ct \leq I(\mathcal{M}(\mathbf{A})) \leq t$  for some constant  $c$ .  $\square$

## References

- [1] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Association for Computational Linguistics (ACL)*, 1996.
- [2] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.
- [4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [5] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.
- [6] M. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, 2015.
- [7] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
- [8] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.
- [9] Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC’07)*, pages 10–23. IEEE, 2007.

- [10] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 77–86. ACM, 2015.
- [11] Sarah R Allen, Ryan ODonnell, and David Witmer. How to refute a random csp. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 689–708. IEEE, 2015.
- [12] Ryuhei Mori and David Witmer. Lower bounds for csp refutation by sdp hierarchies. *arXiv preprint arXiv:1610.03029*, 2016.
- [13] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*, 2015.
- [14] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.
- [15] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [16] D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. In *Conference on Learning Theory (COLT)*, 2009.
- [17] A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden Markov models. In *Conference on Learning Theory (COLT)*, 2012.
- [18] H. Sedghi and A. Anandkumar. Training input-output recurrent neural networks through spectral methods. *arXiv preprint arXiv:1603.00954*, 2016.
- [19] M. Janzamin, H. Sedghi, and A. Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.
- [20] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable bounds for learning some deep representations. In *International Conference on Machine Learning (ICML)*, pages 584–592, 2014.
- [21] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- [22] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Trans. Information Theory*, 44, 1998.
- [23] P.D. Grunwald. A tutorial introduction to the minimum description length principle. *Advances in MDL: Theory and Applications*, 2005.
- [24] A. Dawid. Statistical theory: The prequential approach. *J. Royal Statistical Society*, 1984.
- [25] Y. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23, 1987.
- [26] K. S. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3), 2001.
- [27] D. P. Foster. Prediction in the worst case. *Annals of Statistics*, 19, 1991.



- [28] M. Opper and D. Haussler. Worst case prediction over sequences under log loss. *The Mathematics of Information Coding, Extraction and Distribution*, 1998.
- [29] Nicolo Cesa-Bianchi and Gabor Lugosi. Worst-case bounds for the logarithmic loss of predictors. *Machine Learning*, 43, 2001.
- [30] V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69, 2001.
- [31] S. M. Kakade and A. Y. Ng. Online bounds for bayesian algorithms. *Proceedings of Neural Information Processing Systems*, 2004.
- [32] M. W. Seeger, S. M. Kakade, and D. P. Foster. Worst-case bounds for some non-parametric bayesian methods, 2005.
- [33] B. S. Clarke and A. R. Barron. Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, 36(3):453–471, 1990.
- [34] David Haussler and Manfred Opper. Mutual information, metric entropy and cumulative relative entropy risk. *ANNALS OF STATISTICS*, 25(6):2451–2492, 1997.
- [35] A. Barron. Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. In Bernardo, Berger, Dawid, and Smith, editors, *Bayesian Statistics 6*, pages 27–52, 1998.
- [36] A. Barron, M. Schervish, and L. Wasserman. The consistency of posterior distributions in nonparametric problems. *ANNALS OF STATISTICS*, 2(27):536–561, 1999.
- [37] P. Diaconis and D. Freedman. On the consistency of Bayes estimates. *ANNALS OF STATISTICS*, 14:1–26, 1986.
- [38] T. Zhang. Learning bounds for a generalized family of Bayesian posterior distributions. *Proceedings of Neural Information Processing Systems*, 2006.
- [39] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theor.*, 1978.
- [40] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 655–664. ACM, 2013.
- [41] Ryan ODonnell and David Witmer. Goldreich’s prg: Evidence for near-optimal polynomial stretch. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 1–12. IEEE, 2014.
- [42] Siu On Chan, James R Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large lp relaxations. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 350–359. IEEE, 2013.
- [43] Boaz Barak, Siu On Chan, and Pravesh K Kothari. Sum of squares lower bounds from pairwise independence. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 97–106. ACM, 2015.
- [44] Siavosh Benabbas, Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. Sdp gaps from pairwise independence. *Theory of Computing*, 8(1):269–289, 2012.

- [45] Madhur Tulsiani and Pratik Worah. Ls+ lower bounds from pairwise independence. In *2013 IEEE Conference on Computational Complexity*, pages 121–132. IEEE, 2013.
- [46] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnfs. *CoRR*, *abs/1404.3378*, 1(2.1):2–1, 2014.
- [47] Amit Daniely. Complexity theoretic limitations on learning halfspaces. *arXiv preprint arXiv:1505.05800*, 2015.